



**TUGAS AKHIR - KI141502**

# **RANCANG BANGUN APLIKASI EDUKASI INTERAKTIF UNTUK MELATIH BERPIKIR KOMPUTASIONAL MATERI GRAF**

**ERIKO AGNITA MASROERI**  
**NRP 5113100122**

**Dosen Pembimbing**  
**Rully Soelaiman, S.Kom., M.Kom.**  
**Fajar Baskoro, S.Kom., M.T.**

**DEPARTEMEN TEKNIK INFORMATIKA**  
**Fakultas Teknologi Informasi**  
**Institut Teknologi Sepuluh Nopember**  
**Surabaya 2017**

*[Halaman ini sengaja dikosongkan]*



**TUGAS AKHIR - KI141502**

# **RANCANG BANGUN APLIKASI EDUKASI INTERAKTIF UNTUK MELATIH BERPIKIR KOMPUTASIONAL MATERI GRAF**

**ERIKO AGNITA MASROERI  
NRP 5113100122**

**Dosen Pembimbing  
Rully Soelaiman, S.Kom., M.Kom.  
Fajar Baskoro, S.Kom., M.T.**

**DEPARTEMEN TEKNIK INFORMATIKA  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017**

*[Halaman Ini Sengaja Dikosongkan]*



**FINAL PROJECT- KI141502**

# **DESIGN OF INTERACTIVE EDUCATIONAL APPLICATION FOR LEARNING COMPUTATIONAL THINKING GRAPH MODULE**

**ERIKO AGNITA MASROERI  
NRP 5113100122**

**Advisor  
Rully Soelaiman, S.Kom., M.Kom.  
Fajar Baskoro, S.Kom., M.T.**

**DEPARTMENT OF INFORMATICS  
Faculty of Information Technology  
Sepuluh Nopember Institute of Technology  
Surabaya 2017**

*[Halaman ini sengaja dikosongkan]*

## **LEMBAR PENGESAHAN**

### **RANCANG BANGUN APLIKASI EDUKASI INTERAKTIF UNTUK MELATIH BERPIKIR KOMPUTASIONAL MATERI GRAF**

### **TUGAS AKHIR**

**Diajukan Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada  
Rumpun Mata Kuliah Algoritma Pemrograman  
Program Studi S-1 Departemen Teknik Informatika  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember**

Oleh:

**ERIKO AGNITA MASROERI  
NRP. 5113 100 122**

**Disetujui oleh Dosen Pembimbing Tugas Akhir:  
Rully Soelaiman, S.Kom., M.Kom.  
NIP: 197002131994021001**

**Fajar Baskoro, S.Kom., M.T.  
NIP: 197404031999031002**



(pembimbing 1)

(pembimbing 2)

**SURABAYA  
JULI, 2017**

*[Halaman ini sengaja dikosongkan]*



# **RANCANG BANGUN APLIKASI EDUKASI INTERAKTIF UNTUK MELATIH BERPIKIR KOMPUTASIONAL MATERI GRAF**

<b>Nama Mahasiswa</b>	<b>: Eriko Agnita Masroeri</b>
<b>NRP</b>	<b>: 5113100122</b>
<b>Jurusan</b>	<b>: Teknik Informatika FTIf-ITS</b>
<b>Dosen Pembimbing I</b>	<b>: Rully Soelaiman, S.Kom., M.Kom.</b>
<b>Dosen Pembimbing II</b>	<b>: Fajar Baskoro, S.Kom., M.T.</b>

## **Abstrak**

*Cara berpikir sangat berpengaruh dalam menyelesaikan sebuah masalah. Berpikir komputasional adalah metode untuk menyelesaikan sebuah masalah dengan cara berpikir secara logika informatika. Berpikir komputasional bukan berarti berpikir seperti komputer, melainkan berpikir tentang komputasi di mana seseorang dituntut untuk memformulasikan masalah dalam bentuk masalah komputasi dan menyusun solusi komputasi yang baik.*

*Bebras merupakan sebuah inisiatif yang bertujuan untuk mengenalkan cara berpikir komputasional pada pelajar serta masyarakat luas. Terdapat kompetisi Bebras yang berisi kumpulan soal yang dilengkapi dengan berbagai gambar menarik agar memotivasi serta memudahkan peserta dalam memahami dan mengerjakan soal. Dalam melakukan kompetisi, peserta harus memberikan solusi untuk setiap persoalan yang ada tanpa harus memiliki pengetahuan tentang informatika terlebih dahulu. Soal-soal yang diberikan di dalam kompetisi Bebras ini menerapkan aspek ilmu komputasi atau informatika yang dimaksudkan untuk melatih berpikir secara logis, kreatif dan kritis serta dapat menerapkan konsep berpikir komputasional untuk menyelesaikan masalah yang diberikan.*

***Kata kunci: Berpikir komputasional, Bebras, kompetisi Bebras***

*[Halaman ini sengaja dikosongkan]*

# **DESIGN OF INTERACTIVE EDUCATIONAL APPLICATION FOR LEARNING COMPUTATIONAL THINKING GRAPH MODULE**

**Student Name** : Eriko Agnita Masroeri  
**NRP** : 5113100122  
**Major** : Informatics Department FTIf-ITS  
**Advisor I** : Rully Soelaiman, S.Kom., M.Kom.  
**Advisor II** : Fajar Baskoro, S.Kom., M.T.

## **Abstract**

*Way of thinking is very influential in solving a problem. Computational thinking is a method to solve a problem by way of logical informatics thinking. Computational thinking does not mean thinking like a computer, but rather thinking about computing where someone is required to formulate a problem in the form of computing problems and compose a good computing solution,*

*Bebras is an initiative that aims to introduce the way of computational thinking to the students and the wider community. There is a Bebras competition that contains a collection of questions equipped with various interesting images to motivate and ease participants in understanding and working on the problem. In the competition, participants must provide solutions to any existing problems without having knowledge of informatics first. The questions given in this Bebras competition apply aspects of computational science or informatics that are intended to train logical, creative and critical thinking and can applying the concept of computational thinking to solve a given problem.*

**Keywords:** *Computational thinking, Bebras, Bebras competition.*

*[Halaman ini sengaja dikosongkan]*

## KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Segala puji bagi Allah SWT, Tuhan semesta alam yang telah melimpahkan rahmat dan hidayah-Nya. Sholawat serta salam tak lupa tercurah kepada Nabi Muhammad Shollallahu ‘Alaihi Wassalam.

Dalam pelaksanaan dan pembuatan tugas akhir ini, penulis menerima banyak bantuan dari berbagai pihak, oleh karena itu melalui lembaran kata pengantar ini, penulis menyampaikan ucapan terima kasih kepada:

1. Allah SWT atas segala rahmat, karunia, kemudahan, dan hidayahnya yang telah diberikan selama ini.
2. Kedua orang tua dan saudara penulis, yang selalu mencurahkan doa, dukungan semangat, perhatian, serta kasih sayang kepada penulis.
3. Bapak Rully Soelaiman, S.Kom., M.Kom. selaku dosen pembimbing I.
4. Bapak Fajar Baskoro, S.Kom., M.T. selaku dosen pembimbing II.
5. Teman-teman TC13 yang telah memberikan semangat.
6. Para admin laboratorium IGS (Interaksi Grafika dan Seni) dan RPL (Rekayasa Perangkat Lunak) sebagai tempat tukar ilmu dan pengalaman selama pengerjaan tugas akhir.
7. Sahabat-sahabati Sepuluh Nopember yang menemani penulis dalam hal semangat spiritual.
8. Juga tidak lupa kepada semua pihak yang belum sempat disebutkan satu per satu yang telah membantu terselesaikannya tugas akhir ini, penulis mengucapkan terima kasih.

Penulis telah mengerjakan sebaik mungkin dalam penyusunan aplikasi tugas akhir ini. Penulis juga memohon maaf apabila terdapat kekurangan, kesalahan, maupun kelalaian yang telah penulis lakukan.

Surabaya, Juli 2017

Eriko Agnita Masroeri

## DAFTAR ISI

LEMBAR PENGESAHAN.....	v
Abstrak .....	vii
Abstract .....	ix
KATA PENGANTAR.....	xi
DAFTAR ISI .....	xiii
DAFTAR GAMBAR .....	xvii
DAFTAR TABEL .....	xxi
DAFTAR KODE SUMBER .....	xxiii
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	2
1.5 Manfaat.....	2
1.6 Metodologi .....	3
1.7 Sistematika Penulisan .....	4
BAB II TINJAUAN PUSTAKA.....	7
2.1 Kompetisi Bebras .....	7
2.2 Berpikir Komputasional .....	7
2.3 WebGL .....	8
2.4 Graf.....	9
2.5 <i>Shortest Path</i> .....	10
2.6 <i>Decision Tree</i> .....	10
2.7 Unity .....	11
2.8 Pemrograman C# .....	11
2.9 HTML5.....	12
2.10 Visual Studio .....	12
BAB III ANALISIS DAN PERANCANGAN.....	15
3.1 Analisis Sistem .....	15
3.1.1 Analisis Permasalahan .....	15
3.1.2 Deskripsi Umum Sistem .....	16
3.1.3 Spesifikasi Kebutuhan Perangkat Lunak .....	16
3.2 Perancangan.....	31

3.2.1	Lingkungan Perancangan Perangkat Lunak.....	32
3.2.2	Perancangan Arsitektur Sistem .....	32
3.2.3	Perancangan Basis Data .....	33
3.2.4	Perancangan Antarmuka Pengguna .....	37
BAB IV IMPLEMENTASI.....		47
4.1	Lingkungan Implementasi Perangkat Lunak .....	47
4.2	Implementasi Antarmuka Pengguna .....	48
4.2.1	Implementasi Antarmuka Halaman Register Pengguna .....	48
4.2.2	Implementasi Antarmuka Halaman Login Pengguna .....	53
4.2.3	Implementasi Antarmuka Halaman Beranda .....	58
4.2.4	Implementasi Antarmuka Halaman Kompetisi .....	58
4.2.5	Implementasi Antarmuka Halaman Detail Peserta .....	59
4.2.6	Implementasi Halaman Antarmuka Tentang Bebras.....	62
4.2.7	Implementasi Antarmuka Halaman Penjelasan Soal.....	63
4.2.8	Implementasi Antarmuka Halaman Selesai .....	63
4.2.9	Implementasi Antarmuka Halaman Peringkat .....	67
4.3	Implementasi Pembuatan Aplikasi .....	69
4.3.1	Implementasi Pembuatan <i>Scene</i> .....	69
4.3.2	Implementasi <i>Import Assets</i> .....	70
4.3.3	Implementasi <i>Load</i> Objek ke <i>Scene</i> .....	71
4.3.4	Implementasi Pembuatan <i>Script</i> .....	71
4.3.5	Implementasi Pembuatan Soal .....	72
BAB V PENGUJIAN DAN EVALUASI .....		108
5.1	Lingkungan Uji Coba .....	109
5.2	Pengujian Fungsionalitas .....	109
5.2.1	Pengujian Fungsionalitas Melihat Tentang Bebras .....	110
5.2.2	Pengujian Fungsionalitas Melihat Detail Peserta.....	112
5.2.3	Pengujian Fungsionalitas Mengerjakan Soal .....	113
5.2.4	Pengujian Fungsionalitas Melihat Nilai .....	118
5.2.5	Pengujian Fungsionalitas Melihat Peringkat.....	120
5.3	Evaluasi Pengujian.....	121
BAB VI KESIMPULAN DAN SARAN .....		123
6.1.	Kesimpulan .....	123



6.2. Saran.....	123
DAFTAR PUSTAKA.....	125
LAMPIRAN A PANDUAN PENGGUNAAN APLIKASI.....	127
LAMPIRAN B PANDUAN PEMBUATAN SOAL.....	137
LAMPIRAN C KUISONER PESERTA .....	157
BIODATA PENULIS.....	161

*[Halaman ini sengaja dikosongkan]*

## DAFTAR GAMBAR

Gambar 2.1 Suatu <i>graph</i> dengan 6 simpul dan 7 sisi.....	9
Gambar 2.2 Tampilan Antarmuka Visual Studio .....	13
Gambar 3.1 DFD Level 0 .....	18
Gambar 3.2 DFD Level 1 untuk Proses Registrasi.....	18
Gambar 3.3 DFD Level 1 .....	18
Gambar 3.4 Diagram Kasus Penggunaan .....	20
Gambar 3.5 Diagram Aktivitas Melihat Tentang Bebras .....	21
Gambar 3.6 Diagram Aktivitas Melihat Detail Peserta.....	23
Gambar 3.7 Diagram <i>Sequence</i> Melihat Detail Peserta .....	24
Gambar 3.8 Diagram <i>Robustness</i> Melihat Detail Peserta .....	24
Gambar 3.9 Diagram Aktivitas Mengerjakan Soal .....	26
Gambar 3.10 Diagram <i>Sequence</i> Mengerjakan Soal .....	27
Gambar 3.11 Diagram <i>Robustness</i> Mengerjakan Soal .....	27
Gambar 3.12 Diagram Aktivitas Melihat Nilai .....	28
Gambar 3.13 Diagram <i>Sequence</i> Melihat Nilai.....	29
Gambar 3.14 Diagram <i>Robustness</i> Melihat Nilai.....	29
Gambar 3.15 Diagram Aktivitas Melihat Peringkat.....	30
Gambar 3.16 Diagram <i>Sequence</i> Melihat Peringkat .....	31
Gambar 3.17 Diagram <i>Robustness</i> Melihat Peringkat .....	31
Gambar 3.18 Arsitektur Sistem .....	32
Gambar 3.19 <i>Conceptual Data Model</i> .....	35
Gambar 3.20 <i>Physical Data Model</i> .....	36
Gambar 3.21 Rancangan Antarmuka Halaman Register.....	38
Gambar 3.22 Rancangan Antarmuka Halaman Login .....	38
Gambar 3.23 Rancangan Antarmuka Halaman Beranda.....	40
Gambar 3.24 Rancangan Antarmuka Halaman Kompetisi .....	41
Gambar 3.25 Rancangan Antarmuka Halaman Detail Peserta....	42
Gambar 3.26 Rancangan Antarmuka Halaman Tentang Bebras.	42
Gambar 3.27 Rancangan Antarmuka Halaman Penjelasan Soal.	43
Gambar 3.28 Rancangan Antarmuka Halaman Soal.....	44
Gambar 3.29 Rancangan Antarmuka Halaman Selesai.....	45
Gambar 3.30 Rancangan Antarmuka Halaman Peringkat.....	46

Gambar 4.1 Implementasi Antarmuka Halaman Register Pengguna .....	48
Gambar 4.2 Implementasi Antarmuka Halaman Login Pengguna .....	54
Gambar 4.3 Implementasi Halaman Antarmuka Beranda.....	58
Gambar 4.4 Implementasi Halaman Antarmuka Kompetisi.....	58
Gambar 4.5 Implementasi Halaman Antarmuka Detail Peserta..	59
Gambar 4.6 Implementasi Halaman Antarmuka Tentang Bebras .....	63
Gambar 4.7 Implementasi Halaman Antarmuka Penjelasan Soal .....	63
Gambar 4.8 Implementasi Antarmuka Halaman Selesai.....	64
Gambar 4.9 Implementasi Antarmuka Halaman Peringkat.....	67
Gambar 4.10 Pembuatan <i>Scene</i> .....	70
Gambar 4.11 <i>Import Asset</i> ke Projek Unity .....	70
Gambar 4.12 <i>Load Objek</i> ke <i>Scene</i> .....	71
Gambar 4.13 Pembuatan <i>Script</i> Baru .....	72
Gambar 5.1 Halaman Beranda.....	111
Gambar 5.2 Halaman Tentang Bebras.....	111
Gambar 5.3 Halaman Beranda.....	112
Gambar 5.4 Halaman Detail Peserta.....	113
Gambar 5.5 Halaman Penjelasan Soal.....	114
Gambar 5.6 Halaman Soal 1 .....	114
Gambar 5.7 Halaman Soal 2.....	115
Gambar 5.8 Halaman Soal 3.....	115
Gambar 5.9 Halaman Soal 4.....	116
Gambar 5.10 Halaman Soal 1 yang Telah Dijawab .....	116
Gambar 5.11 Halaman Soal 2 yang Telah Dijawab .....	117
Gambar 5.12 Halaman Soal 3 yang Telah Dijawab .....	117
Gambar 5.13 Halaman Soal 4 yang Telah Dijawab .....	118
Gambar 5.14 Halaman Soal.....	119
Gambar 5.15 Halaman Selesai.....	119
Gambar 5.16 Halaman Selesai.....	120
Gambar 5.17 Halaman Peringkat.....	121
Gambar A.7.1 Halaman Beranda.....	127

Gambar A.7.2 Halaman Detail Peserta .....	128
Gambar A.7.3 Halaman Tentang Bebras.....	129
Gambar A.7.4 Halaman Kompetisi .....	130
Gambar A.7.5 Halaman Login .....	131
Gambar A.7.6 Halaman Register .....	132
Gambar A.7.7 Halaman Penjelasan Soal.....	133
Gambar A.7.8 Halaman Soal.....	134
Gambar A.7.9 Halaman Selesai .....	135
Gambar A.7.10 Halaman Peringkat.....	136
Gambar B.8.1 Tampilan Awal Unity .....	137
Gambar B.8.2 Membuat Panel .....	139
Gambar B.8.3 Memberi Warna pada Panel.....	140
Gambar B.8.4 Hasil Panel yang Telah Diberi Warna .....	140
Gambar B.8.5 Membuat <i>Text</i> .....	141
Gambar B.8.6 Implementasi Pembuatan <i>Text</i> .....	141
Gambar B.8.7 <i>Input</i> Gambar .....	142
Gambar B.8.8 Tipe Gambar .....	142
Gambar B.8.9 Membuat Gambar .....	143
Gambar B.8.10 <i>Drag and Drop</i> Gambar.....	143
Gambar B.8.11 Implementasi Pembuatan Gambar .....	144
Gambar B.8.12 Membuat Tombol .....	144
Gambar B.8.13 Implementasi Tombol .....	145
Gambar B.8.14 Hasil Implementasi Tombol.....	145
Gambar B.8.15 Implementasi <i>Text</i> pada Jalan .....	146
Gambar B.8.16 Hasil Implementasi Soal 1 Secara Keseluruhan .....	146
Gambar B.8.17 Hasil Implementasi Pembuatan Teks.....	147
Gambar B.8.18 Hasil Implementasi Pembuatan Gambar Simpul dan Sisi .....	148
Gambar B.8.19 Hasil Implementasi Pembuatan Teks pada Sisi	148
Gambar B.8.20 Hasil Implementasi Gambar Semua Simpul, Sisi, dan Teks pada Sisi.....	149
Gambar B.8.21 Implementasi Pembuatan Tombol .....	149
Gambar B.8.22 Hasil Implementasi Pembuatan Tombol .....	150

Gambar B.8.23 Hasil Implementasi Soal 2 Secara Keseluruhan .....	150
Gambar B.8.24 Hasil Implementasi Soal 3 Secara Keseluruhan .....	151
Gambar B.8.25 Hasil Implementasi Teks dan Gambar .....	152
Gambar B.8.26 Implementasi Tombol .....	152
Gambar B.8.27 Hasil Implementasi Tombol .....	153
Gambar B.8.28 Hasil Implementasi Soal 4 Secara Keseluruhan .....	153
Gambar B.8.29 Pembuatan <i>Script</i> .....	154
Gambar B.8.30 <i>Reference Object</i> pada <i>Script</i> .....	155
Gambar C.9.1 Kuisoner Putri .....	157
Gambar C.9.2 Kuisoner Hasanudin .....	158
Gambar C.9.3 Kuisoner Reviana .....	159

## DAFTAR TABEL

Tabel 3.1 Kebutuhan Fungsional.....	17
Tabel 3.2 Kualitas Perangkat Lunak .....	19
Tabel 3.3 Kasus Penggunaan.....	21
Tabel 3.4 Spesifikasi Kasus Penggunaan UC-0001 .....	22
Tabel 3.5 Spesifikasi Kasus Penggunaan UC-0002 .....	23
Tabel 3.6 Spesifikasi Kasus Penggunaan UC-0003 (1).....	25
Tabel 3.7 Spesifikasi Kasus Penggunaan UC-0003 (2).....	26
Tabel 3.8 Spesifikasi Kasus Penggunaan UC-0004 .....	28
Tabel 3.9 Spesifikasi Kasus Penggunaan UC-0005 .....	30
Tabel 3.10 Lingkungan Perancangan Perangkat Lunak .....	32
Tabel 3.11 Atribut Tabel Peserta.....	33
Tabel 3.12 Atribut Tabel Soal .....	34
Tabel 3.13 Atribut Tabel Jawab .....	34
Tabel 3.14 Atribut Antarmuka Halaman Register Pengguna .....	37
Tabel 3.15 Atribut Antarmuka Halaman Login Pengguna.....	39
Tabel 3.16 Atribut Antarmuka Halaman Beranda.....	39
Tabel 3.17 Atribut Antarmuka Halaman Kompetisi .....	40
Tabel 3.18 Atribut Antarmuka Halaman Detail Peserta.....	41
Tabel 3.19 Atribut Antarmuka Halaman Tentang Bebras.....	42
Tabel 3.20 Atribut Antarmuka Halaman Penjelasan Soal .....	43
Tabel 3.21 Atribut Antarmuka Halaman Soal .....	44
Tabel 3.22 Atribut Antarmuka Halaman Selesai.....	45
Tabel 3.23 Atribut Antarmuka Halaman Peringkat.....	46
Tabel 4.1 Lingkungan Implementasi Perangkat Lunak.....	47
Tabel 5.1 Lingkungan Pengujian Fungsionalitas Perangkat Lunak .....	109
Tabel 5.2 Skenario Uji Coba Fungsionalitas .....	110
Tabel 5.3 Uji Coba Melihat Tentang Bebras.....	110
Tabel 5.4 Uji Coba Melihat Detail Peserta.....	112
Tabel 5.5 Uji Coba Mengerjakan Soal .....	113
Tabel 5.6 Uji Coba Melihat Nilai .....	118
Tabel 5.7 Uji Coba Melihat Peringkat.....	120
Tabel 5.8 Rekapitulasi Hasil Uji Fungsionalitas .....	121

Tabel C.9.1 Hasil Kuisioner Peserta.....160



## DAFTAR KODE SUMBER

Kode Sumber 4.1 Fungsi Register Pengguna .....	51
Kode Sumber 4.2 Kontrol Register Pengguna .....	53
Kode Sumber 4.3 Fungsi Login Pengguna .....	56
Kode Sumber 4.4 Kontrol Login Pengguna .....	57
Kode Sumber 4.5 Fungsi Detail Peserta .....	60
Kode Sumber 4.6 Kontrol Detail Peserta .....	61
Kode Sumber 4.7 Kontrol Jumlah Peserta .....	62
Kode Sumber 4.8 Fungsi Selesai .....	65
Kode Sumber 4.9 Kontrol Fungsi Selesai .....	67
Kode Sumber 4.10 Fungsi Peringkat .....	68
Kode Sumber 4.11 Kontrol Fungsi Peringkat .....	69
Kode Sumber 4.12 Inisiasi Variabel Jalan .....	73
Kode Sumber 4.13 Fungsi Jalan 1 .....	73
Kode Sumber 4.14 Fungsi Jalan 2 .....	74
Kode Sumber 4.15 Fungsi Jalan 3 .....	75
Kode Sumber 4.16 Fungsi Jalan 4 .....	75
Kode Sumber 4.17 Fungsi Jalan 5 .....	76
Kode Sumber 4.18 Fungsi Jalan 6 .....	76
Kode Sumber 4.19 Fungsi Jalan 7 .....	77
Kode Sumber 4.20 Fungsi Jalan 8 .....	78
Kode Sumber 4.21 Fungsi Jalan 9 .....	78
Kode Sumber 4.22 Fungsi Jalan 10 .....	79
Kode Sumber 4.23 Fungsi Jalan 11 .....	79
Kode Sumber 4.24 Fungsi Jalan 12 .....	80
Kode Sumber 4.25 Fungsi Jalan 13 .....	80
Kode Sumber 4.26 Fungsi Jalan 14 .....	81
Kode Sumber 4.27 Fungsi Jalan 15 .....	82
Kode Sumber 4.28 Fungsi Jalan 16 .....	82
Kode Sumber 4.29 Fungsi Jalan 17 .....	83
Kode Sumber 4.30 Fungsi Jalan 18 .....	83
Kode Sumber 4.31 Fungsi Jalan 19 .....	84
Kode Sumber 4.32 Fungsi Jalan 20 .....	84
Kode Sumber 4.33 Fungsi Jalan 21 .....	85

Kode Sumber 4.34 Fungsi Jalan 22 .....	86
Kode Sumber 4.35 Fungsi Jalan 23 .....	86
Kode Sumber 4.36 Fungsi Jalan 24 .....	87
Kode Sumber 4.37 Fungsi Jalan 25 .....	87
Kode Sumber 4.38 Fungsi Jalan 26 .....	88
Kode Sumber 4.39 Fungsi Jalan 27 .....	88
Kode Sumber 4.40 Inisiasi Variabel Rumah .....	89
Kode Sumber 4.41 Fungsi Rumah 1 .....	89
Kode Sumber 4.42 Fungsi Rumah 2 .....	90
Kode Sumber 4.43 Fungsi Rumah 3 .....	90
Kode Sumber 4.44 Inisiasi Variabel Jalan .....	91
Kode Sumber 4.45 Fungsi Jalan 1 .....	92
Kode Sumber 4.46 Fungsi Jalan 2 .....	92
Kode Sumber 4.47 Fungsi Jalan 3 .....	93
Kode Sumber 4.48 Fungsi Jalan 4 .....	93
Kode Sumber 4.49 Fungsi Jalan 5 .....	94
Kode Sumber 4.50 Fungsi Jalan 6 .....	94
Kode Sumber 4.51 Fungsi Jalan 7 .....	95
Kode Sumber 4.52 Fungsi Jalan 8 .....	96
Kode Sumber 4.53 Fungsi Jalan 9 .....	96
Kode Sumber 4.54 Inisiasi Variabel Satuan Kilometer .....	97
Kode Sumber 4.55 Fungsi Tombol 1 .....	97
Kode Sumber 4.56 Fungsi Tombol 2 .....	98
Kode Sumber 4.57 Fungsi Tombol 3 .....	98
Kode Sumber 4.58 Inisiasi Variabel Jumlah Halaman Soal .....	99
Kode Sumber 4.59 Implementasi <i>Random</i> Soal .....	101
Kode Sumber 4.60 Implementasi <i>Insert</i> Nilai .....	101
Kode Sumber 4.61 Kontrol <i>Insert</i> Nilai .....	102
Kode Sumber 4.62 Implementasi Waktu Hitung Mundur .....	104
Kode Sumber 4.63 Implementasi Terjemahan Bahasa .....	107

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Kompetisi Bebras didirikan di negara Lithuania oleh Prof. Valentina Dagiene dari University of Vilnius pada tahun 2004. Bebras adalah istilah dalam bahasa Lithuania untuk “beaver” (dalam bahasa Indonesia adalah “berang-berang”). Bebras dipilih sebagai simbol tantangan (*challenge*), karena hewan berang-berang berusaha keras untuk mencapai target secara sempurna dalam aktivitasnya sehari-hari. Mereka membuat bendungan dari ranting-ranting pohon di sungai atau aliran air dan membuat rumahnya sendiri. Kompetisi ini disebut Bebras untuk menunjukkan kerja keras dan kecerdasan diperlukan di dalam kehidupan [1].

Kompetisi Bebras dilaksanakan setiap tahun. Negara yang sudah berpartisipasi mengikuti Bebras ada 50 negara, belum termasuk Indonesia. Pada tahun 2015, jumlah peserta yang mengikuti Bebras mencapai 1,3 juta siswa dari berbagai belahan dunia. Setelah kunjungannya ke Indonesia pada bulan Februari 2016, Prof Valentina Dagiene (Vilnius University, Lithuania) yaitu penggagas (*founder*) Bebras Internasional, mengundang Indonesia menjadi *observer* pada Workshop Internasional Bebras pada bulan Mei tahun 2016.

Pada kompetisi Bebras di Indonesia, peserta diminta untuk menjawab soal-soal terkait dengan pemecahan masalah, berpikir komputasional dan kreativitas. Terdapat tiga kategori pada kompetisi Bebras di Indonesia, yaitu siaga, penggalang dan penegak. Tingkat kesulitan soal disesuaikan dengan kategori dari masing-masing peserta.

Saat ini, kompetisi Bebras Indonesia mempunyai *website* yaitu Olympia.id, dimana *website* tersebut menjadi tempat untuk peserta menjawab soal-soal kompetisi Bebras Indonesia. Namun, pada *website* tersebut masih memiliki masalah terutama pada saat menampilkan soal-soal interaktif pada perangkat bergerak.

Dengan aplikasi ini nantinya, diharapkan dapat mengatasi masalah pada *website* kompetisi Bebras Indonesia, di mana pengguna dapat melihat dan menjawab soal-soal interaktif pada perangkat bergerak dengan lancar.

## 1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam tugas akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana membuat soal interaktif pada materi graf?
2. Bagaimana agar soal-soal interaktif tersebut dapat dibuka dengan baik dan tanpa masalah pada perangkat bergerak?

## 1.3 Batasan Masalah

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan antara lain:

1. Metode soal-soal yang akan dibuat adalah metode soal interaktif.
2. Soal-soal yang akan dibuat adalah modul graf materi *shortest path* dan *decision tree*.
3. *Browser* yang digunakan memiliki spesifikasi untuk WebGL.
4. Pembuat soal mengerti cara menggunakan aplikasi Unity.

## 1.4 Tujuan

Tujuan dari pembuatan tugas akhir ini adalah untuk membangun aplikasi media edukasi untuk berpikir secara komputasional pada materi graf.

## 1.5 Manfaat

Manfaat dari hasil pembuatan tugas akhir ini adalah:

1. Soal-soal yang diberikan dapat membuat peserta mengembangkan cara berpikir secara komputasional.
2. Bentuk soal yang menarik dapat memotivasi peserta untuk menyelesaikannya.

3. Menumbuhkan kebiasaan berpikir secara komputasional sejak dini.

## 1.6 Metodologi

Pembuatan tugas akhir dilakukan menggunakan metodologi sebagai berikut:

### A. Studi literatur

Tahap studi literatur merupakan tahap pembelajaran dan pengumpulan informasi yang digunakan untuk mengimplementasikan tugas akhir. Tahap ini diawali dengan pengumpulan literatur, eksplorasi teknologi dan pustaka, serta pemahaman dasar teori yang digunakan pada topik tugas akhir. Literatur-literatur yang dimaksud disebutkan sebagai berikut:

1. Kompetisi Bebras.
2. Berpikir Komputasional.
3. WebGL.
4. Graf.
5. *Shortest Path*.
6. *Decision Tree*.
7. Unity.
8. Pemrograman C#.
9. HTML5.
10. Visual Studio.

### B. Perancangan perangkat lunak

Pada tahap ini diawali dengan melakukan analisis awal terhadap permasalahan utama yang muncul pada topik tugas akhir. Kemudian dilakukan perancangan perangkat lunak yang meliputi penentuan data yang akan digunakan dan proses-proses yang akan dilaksanakan meliputi perumusan kebutuhan fungsional, kebutuhan non-fungsional, identifikasi pengguna, kasus penggunaan, rancangan antarmuka, pembuatan rancangan basis data.

- C. Implementasi dan pembuatan sistem  
Pada tahap ini dilakukan implementasi antarmuka, aplikasi, dan soal. Manipulasi data menggunakan *database* MySQL. *Asset-asset* seperti gambar untuk pembuatan *game* didapat dari internet. Selanjutnya aplikasi ini dibangun dengan aplikasi Unity.
- D. Uji coba dan evaluasi  
Pada tahap ini dilakukan uji coba dengan menggunakan beberapa macam kondisi untuk mencoba aplikasi bisa berjalan atau tidak. Uji fungsionalitas untuk mengetahui apakah aplikasi sudah memenuhi semua kebutuhan fungsional selain itu juga dilakukan pengujian non-fungsionalitas apakah aplikasi sudah bisa memberi soal-soal yang menantang dan kesan *immersive* kepada pengguna. Pengujian aplikasi ini dilakukan dengan metode kuesioner kepada pengguna yang sebelumnya telah diminta untuk menjalankan aplikasi secara langsung.
- E. Penyusunan laporan tugas akhir  
Pada tahap ini dilakukan penyusunan laporan yang berisi dasar teori, dokumentasi dari perangkat lunak, dan hasil-hasil yang diperoleh selama pengerjaan tugas akhir.

## 1.7 Sistematika Penulisan

Buku tugas akhir ini terdiri dari beberapa bab yang dijelaskan sebagai berikut:

### **BAB I PENDAHULUAN**

Bab ini berisi latar belakang, rumusan dan batasan permasalahan, tujuan dan manfaat pembuatan tugas akhir, metodologi yang digunakan, dan sistematika penyusunan tugas akhir.

### **BAB II TINJAUAN PUSTAKA**

Bab ini membahas dasar pembuatan dan beberapa teori penunjang yang berhubungan dengan pokok

pembahasan yang mendasari pembuatan tugas akhir ini.

### **BAB III ANALISIS DAN PERANCANGAN**

Bab ini membahas analisis dari sistem yang dibuat meliputi analisis permasalahan, deskripsi umum perangkat lunak, dan spesifikasi kebutuhan. Kemudian membahas rancangan dari sistem yang dibuat meliputi lingkungan perancangan, rancangan arsitektur, basis data, dan antarmuka.

### **BAB IV IMPLEMENTASI**

Bab ini membahas implementasi dari rancangan sistem yang dilakukan pada tahap perancangan. Penjelasan implementasi meliputi implementasi antar muka, implementasi pembuatan aplikasi serta implementasi pembuatan soal.

### **BAB V PENGUJIAN DAN EVALUASI**

Bab ini membahas pengujian dari aplikasi yang dibuat dengan melihat keluaran yang dihasilkan oleh aplikasi dan evaluasi untuk mengetahui kemampuan aplikasi.

### **BAB VI PENUTUP**

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan serta saran untuk pengembangan aplikasi.

*[Halaman ini sengaja dikosongkan]*



## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Kompetisi Bebras**

Bebras adalah inisiatif internasional yang bertujuan untuk mempromosikan informatika (ilmu komputer atau komputasi) dan berpikir komputasional di kalangan siswa sekolah di segala usia. Peserta pada umumnya diawasi oleh guru yang dapat mengintegrasikan tantangan Bebras dalam kegiatan mengajar mereka [2].

Kompetisi Bebras adalah sekumpulan soal yang disebut *Bebras task*. *Bebras task* disajikan dalam bentuk uraian persoalan yang dilengkapi dengan gambar yang menarik, sehingga siswa dapat lebih mudah memahami soal [3]. Soal-soal tersebut dapat dijawab tanpa perlu belajar informatika terlebih dahulu, namun soal tersebut sebetulnya terkait pada konsep tertentu dalam informatika dan *computational thinking* [4].

Penyusun soal Bebras berusaha memilih soal yang menarik untuk memotivasi siswa dalam mengidentifikasi persoalan informatika dan berpikir lebih dalam tentang teknologi. Mereka juga ingin menyajikan sebanyak mungkin topik informatika dan pembelajaran komputer.

#### **2.2 Berpikir Komputasional**

Berpikir komputasional (*computational thinking*) adalah sebuah metode pemecahan masalah dengan mengaplikasikan/melibatkan teknik yang digunakan oleh *software engineer* dalam menulis program. Berpikir komputasional tidak berarti berpikir seperti komputer, melainkan berpikir tentang komputasi di mana seseorang dituntut untuk memformulasikan masalah dalam bentuk masalah komputasi dan menyusun solusi komputasi yang baik (dalam bentuk algoritma) atau menjelaskan mengapa tidak ditemukan solusi yang sesuai. Berpikir komputasional merupakan proses pemecahan masalah yang

mencakup sejumlah karakteristik, seperti menganalisa data secara logis serta menciptakan solusi dengan menggunakan serangkaian langkah secara terurut.

Berpikir komputasional ditujukan untuk menyelesaikan masalah, bukan hanya untuk masalah seputar ilmu komputer, melainkan juga untuk menyelesaikan beragam masalah. Contohnya adalah *Machine learning* yang telah mengubah bagaimana ilmu statistika dimanfaatkan. Sedangkan dalam bidang ilmu biologi, data mining (yang merupakan konsep komputasi) dapat melakukan pencarian pada sejumlah besar data untuk menemukan pola-pola. Harapannya adalah struktur data dan algoritma (yang merupakan teknik abstraksi pada ilmu komputer) dapat menggambarkan struktur protein dengan cara yang menjelaskan fungsi-fungsi mereka. Berpikir komputasional adalah teknik pemecahan masalah yang sangat luas wilayah penerapannya. Tidak mengherankan bahwa memiliki kemampuan tersebut adalah sebuah keharusan bagi seseorang yang hidup pada abad ke-21 ini [4].

### 2.3 WebGL

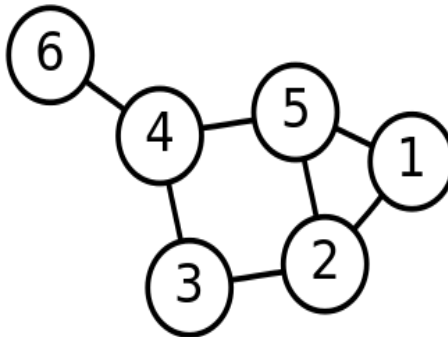
WebGL adalah teknologi *web* yang menyuguhkan akselerasi grafis 3D ke dalam *browser* tanpa memasang perangkat lunak tambahan. WebGL merupakan kependekan dari Web Graphics Library. Untuk API, WebGL pada umumnya dipanggil melalui API Javascript dan penggunaannya selalu melibatkan elemen HTML5 <canvas>.

WebGL pada umunya digunakan untuk desain *web* dan *game* 3D yang berbasis *web*. WebGL juga digunakan oleh beberapa peneliti untuk tujuan sains. Sebagai contoh, dalam buku yang berjudul "Cellular Automata", peneliti menggunakan WebGL untuk mensimulasikan alur Debris. Sedangkan pada *paper* "Visualization of molecular structures using state-of-the-art techniques in WebGL", peneliti menyimulasikan molekul di dalamnya. NASA juga membangun aplikasi *web* yang interaktif

yang disebut Experience Curiosity untuk merayakan ulang tahun ketiga dimana Curiosity Rover mendarat di Mars [5].

## 2.4 Graf

Teori graf atau teori grafik dalam matematika dan ilmu komputer adalah cabang kajian yang mempelajari sifat-sifat graf atau grafik. Ini tidak sama dengan grafika. Secara informal, suatu graf adalah himpunan benda-benda yang disebut simpul (*vertex* atau *node*) yang terhubung oleh sisi (*edge*) atau busur (*arc*). Pada umumnya graf digambarkan sebagai kumpulan titik-titik (melambangkan simpul) yang dihubungkan oleh garis-garis (melambangkan sisi) atau garis berpanah (melambangkan busur). Suatu sisi dapat menghubungkan suatu simpul dengan simpul yang sama. Sisi demikian dinamakan gelang (*loop*) [6].



**Gambar 2.1** Suatu *graph* dengan 6 simpul dan 7 sisi.

Suatu *graph*  $G$  dapat dinyatakan sebagai  $G = \langle V, E \rangle$

Di mana:

$G = \text{Graph}$

$V = \text{Simpul (Vertex atau Node)}$

$E = \text{Sisi (Edge) atau Busur (arc)}$

*Graph G* terdiri atas himpunan  $V$  yang berisikan simpul pada graf tersebut dan himpunan dari  $E$  yang berisi sisi pada graf tersebut. Himpunan  $E$  dinyatakan sebagai pasangan dari simpul yang ada dalam  $V$ .

Contoh definisi dari graf pada gambar 2.1 di atas adalah  $V = \{1, 2, 3, 4, 5, 6\}$  dan  $E = \{(1, 2), (1, 5), (2, 3), (3, 4), (4, 5), (5, 2), (4, 6)\}$ .

## 2.5 Shortest Path

Misalkan  $G = (V, E)$  adalah graf dimana pemetaan  $w : E \rightarrow \mathbb{R}$  didefinisikan. Pasangan  $(G, w)$  disebut dengan jaringan, nilai  $w(e)$  adalah panjang dari sisi  $e$ , sehingga untuk setiap jalan  $W = (e_1, \dots, e_n)$ , panjang  $W$  adalah  $w(W) := w(e_1) + \dots + w(e_n)$ .

Sangat bisa untuk didefinisikan jarak  $d(a, b)$  antara dua simpul  $a$  dan  $b$  di  $G$  sebagai minimum sepanjang panjang jalan dengan simpul awal  $a$  dan simpul akhir  $b$ . Namun, ada kesulitan dengan definisi ini,  $b$  mungkin tidak dapat diakses dari  $a$ . Masalah tersebut diatasi dengan menempatkan  $d(a, b) = \infty$  jika  $b$  tidak dapat diakses dari  $a$ . Setiap jalur  $W$  di mana paling minimum dicapai disebut jalur terpendek (*shortest path*) dari  $a$  ke  $b$  [7].

## 2.6 Decision Tree

Pada setiap langkah, himpunan semua solusi yang mungkin terbagi menjadi dua atau lebih banyak *subset*, yang diwakili oleh cabang disebut pohon keputusan (*decision tree*). Pohon keputusan pada umumnya digunakan dalam klasifikasi. Klasifikasi adalah proses menemukan kumpulan pola atau fungsi-fungsi yang mendeskripsikan dan memisahkan kelas data satu dengan lainnya, untuk dapat digunakan untuk memprediksi data yang belum memiliki kelas data tertentu untuk membantu mengidentifikasi strategi yang paling mungkin mencapai tujuan [7].

Pohon keputusan dibentuk dari 3 tipe dari simpul, yaitu simpul *root*, simpul perantara, dan simpul *leaf*.

- Simpul *root* adalah titik awal dari suatu pohon keputusan.
- Setiap simpul perantara berhubungan dengan suatu pertanyaan atau pengujian.
- Simpul *leaf* memuat suatu keputusan akhir atau kelas target untuk suatu pohon keputusan.

## 2.7 Unity

Unity adalah sebuah *tool* pengembangan *game* yang terintegrasi kuat dengan satu set lengkap alat intuitif dan alur kerja yang cepat untuk membuat 3D interaktif dan konten 2D. Unity merupakan *easy multiplatform publishing*. Unity memiliki *Asset Store* yang menyediakan *asset* untuk diunduh secara gratis maupun berbayar. Terdapat pula *Unity Community* yang menyediakan tutorial secara gratis untuk semua pengguna Unity dalam bahasa Inggris. Fitur *scripting* yang disediakan, mendukung 3 bahasa pemrograman, yaitu JavaScript, C#, dan Boo. *Flexible, easy moving, rotating, dan scaling objects* hanya perlu sebaris kode. Begitu juga dengan *duplicating, removing, dan changing properties*.

Visual Properties Variables yang didefinisikan dengan *scripts* ditampilkan pada Editor. Bisa digeser, *drag and drop*, bisa memilih warna dengan *color picker*. Berbasis .NET. Artinya penjalanan program dilakukan dengan *Open Source .NET platform*, Mono. Serta mendukung pengembangan aplikasi Microsoft, SONY, Qualcomm, BlackBerry, Samsung, Ninetendo, Oculus VR dan Intel [8].

## 2.8 Pemrograman C#

C# merupakan sebuah bahasa pemrograman yang berorientasi objek yang dikembangkan oleh Microsoft sebagai bagian dari inisiatif kerangka .NET Framework. Bahasa pemrograman ini dibuat berbasiskan bahasa C++ yang telah dipengaruhi oleh aspek-aspek ataupun fitur bahasa yang terdapat pada bahasa-bahasa pemrograman lainnya seperti Java, Delphi,

Visual Basic, dan lain-lain) dengan beberapa penyederhanaan. Menurut standar ECMA-334 C# Language Specification, nama C# terdiri atas sebuah huruf Latin C (U+0043) yang diikuti oleh tanda pagar yang menandakan angka # (U+0023). Tanda pagar # yang digunakan memang bukan tanda kres dalam seni musik (U+266F), dan tanda pagar # (U+0023) tersebut digunakan karena karakter kres dalam seni musik tidak terdapat di dalam keyboard standar.

C# didesain untuk memenuhi kebutuhan akan sintaksis C++ yang lebih ringkas dan Rapid Application Development yang tanpa batas (dibandingkan dengan RAD yang terbatas seperti yang terdapat pada Delphi dan Visual Basic).

C# kadang-kadang dapat disebutkan sebagai bahasa pemrograman yang paling mencerminkan dasar dari CLR dimana semua program-program .NET berjalan, dan bahasa ini sangat bergantung pada kerangka tersebut sebab secara spesifik didesain untuk mengambil manfaat dari fitur-fitur yang tersedia pada CLR.

Desain dari *Object Oriented* harus berisi objek-objek yang ada pada problem domain tersebut. Sebagai contoh jika problem domainnya adalah mesin ATM maka objek di dalamnya seharusnya adalah *customer*, rekening, dan lain-lain [9].

## 2.9 HTML5

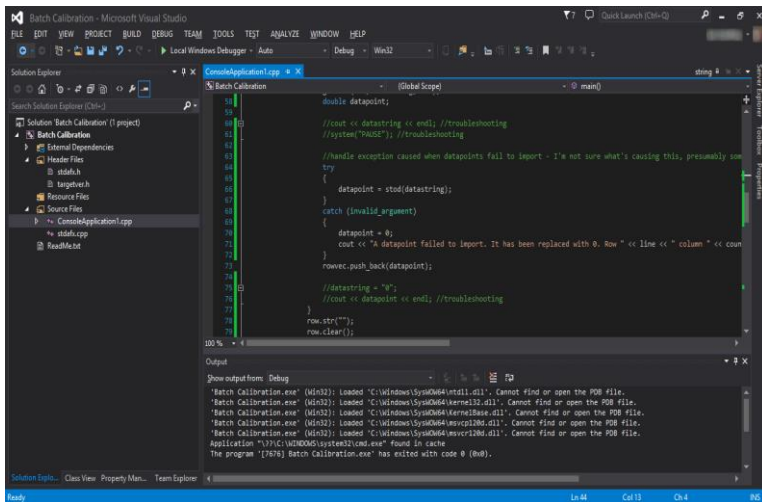
Ada tiga tujuan utama dari HTML5 yaitu:

1. Menentukan perilaku *browser* pada saat bertukar dan menggunakan informasi.
2. Mendefinisikan penanganan kesalahan untuk pertama kali.
3. Mengembangkan bahasa untuk mempermudah penulisan pada aplikasi *web* [10].

## 2.10 Visual Studio

Microsoft Visual Studio merupakan sebuah aplikasi yang terdiri dari *compiler*, SDK, *Integrated Development Environment (IDE)*, dan dokumentasi yang biasa digunakan untuk

mengembangkan aplikasi personal, aplikasi *web*, aplikasi *mobile*, aplikasi *console* maupun komponen aplikasi pada Microsoft Visual Studio. Banyak *compiler* yang disediakan oleh Microsoft Visual Studio, misalnya saja Visual C++, Visual C#, Visual Basic, Visual Basic .NET, Visual InterDev, Visual J++, Visual J#, Visual FoxPro, dan Visual SourceSafe. Banyak pengembang aplikasi yang mengembangkan aplikasinya menggunakan Microsoft Visual Studio karena banyak fungsi yang telah disediakan oleh Microsoft Visual Studio, sehingga pengembang lebih mudah dan sangat terbantu dalam mengembangkan maupun membuat aplikasi. Tampilan antarmuka aplikasi ini dapat dilihat pada Gambar 2.3 [11].



**Gambar 2.2 Tampilan Antarmuka Visual Studio**

*[Halaman ini sengaja dikosongkan]*



## **BAB III**

### **ANALISIS DAN PERANCANGAN**

Bab ini membahas analisis kebutuhan dan rancangan yang akan digunakan untuk membangun perangkat lunak yang diajukan sebagai tugas akhir.

#### **3.1 Analisis Sistem**

Tahap analisis dibagi menjadi beberapa bagian, antara lain analisis permasalahan, deskripsi umum sistem, dan spesifikasi kebutuhan perangkat lunak.

##### **3.1.1 Analisis Permasalahan**

Bebras merupakan sebuah inisiatif untuk memperkenalkan cara berpikir komputasional khususnya kepada para pelajar dan masyarakat luas. Kompetisi yang terdapat pada Bebras menyajikan soal-soal yang menerapkan aspek ilmu komputasi atau informatika yang dimaksudkan agar peserta dapat melatih berpikir secara logis, kreatif, dan kritis dengan menerapkan konsep berpikir komputasional untuk menyelesaikan soal-soal yang diberikan.

Setiap peserta dapat mengerjakan soal-soal secara interaktif pada sistem. Soal-soal yang diberikan dilengkapi dengan berbagai gambar menarik agar memotivasi serta memudahkan peserta dalam memahami dan mengerjakan soal. Peserta dapat menjawab soal secara interaktif sesuai dengan pertanyaan yang telah diberikan. Setiap soal memiliki batasan waktu yang berbeda-beda. Pada saat kompetisi Bebras berlangsung terdapat alat pengatur waktu yang akan dihitung secara mundur sesuai dengan batasan waktu yang telah ditentukan. Nilai peserta akan menentukan peringkat yang akan didapatkan.

### 3.1.2 Deskripsi Umum Sistem

Pada tampilan halaman *web* terdapat tentang Bebras dan detail peserta yang dapat dilihat oleh para peserta. Tentang Bebras berisi informasi mengenai Bebras dan detail peserta berisi informasi mengenai peserta secara rinci. Para peserta dapat melakukan kompetisi Bebras dengan melakukan registrasi dan *login* terlebih dahulu, kemudian peserta dapat mengerjakan soal-soal di dalam kompetisi Bebras secara *online*.

Soal-soal yang diberikan memiliki urutan nomor secara acak. Peserta dapat menjawab soal secara interaktif sesuai dengan pertanyaan yang telah diberikan. Terdapat dua bahasa yang tersedia ketika mengerjakan soal yaitu bahasa Indonesia dan bahasa Inggris, peserta dapat memilih salah satu bahasa yang telah tersedia tersebut. Setiap soal memiliki batasan waktu yang berbeda-beda. Pada saat kompetisi Bebras berlangsung terdapat alat pengatur waktu yang akan dihitung secara mundur sesuai dengan batasan waktu yang telah ditentukan. Peserta yang telah menyelesaikan semua soal dapat melihat nilai serta peringkat yang didapatkan.

### 3.1.3 Spesifikasi Kebutuhan Perangkat Lunak

Sesuai dengan uraian perangkat lunak yang telah dijelaskan pada bagian deskripsi umum sistem, dibutuhkan adanya spesifikasi perangkat lunak agar dapat memberikan solusi dari permasalahan yang diberikan dan dapat bekerja dengan baik dalam mengakomodasi kebutuhan. Spesifikasi kebutuhan perangkat lunak pada tugas akhir ini terdiri dari kebutuhan fungsional yang dapat dilihat pada Tabel 3.1, kebutuhan non-fungsional, aktor, dan kasus penggunaan.

### 3.1.3.1 Kebutuhan Fungsional

**Tabel 3.1 Kebutuhan Fungsional**

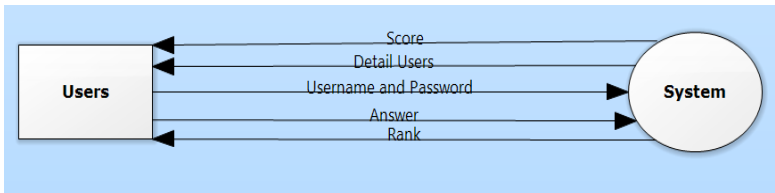
No	Kebutuhan Fungsional	Deskripsi
1	Melihat tentang Bebras	Menampilkan informasi mengenai Bebras
2	Melihat detail peserta	Menampilkan data peserta dan total peserta
3	Menjalankan aktivitas mengerjakan soal	Menjalankan setiap aktivitas mengerjakan soal untuk peserta
4	Melihat nilai	Menampilkan nilai yang didapatkan peserta
5	Melihat peringkat	Menampilkan peringkat yang diperoleh peserta

### 3.1.3.2 Aliran Informasi

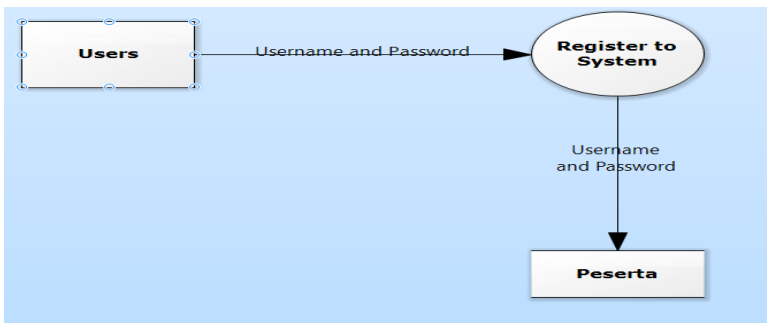
Aliran Informasi untuk sistem dapat dilihat pada Gambar 3.1 berupa *Data Flow Diagram* (DFD) level 0. Dalam diagram tersebut dapat dilihat bahwa pengguna dapat mengakses aplikasi dengan memasukkan data-data yang diperlukan untuk *login* terlebih dahulu. DFD level 0 dapat dilihat pada Gambar 3.1.

### 3.1.3.3 DFD Level 1

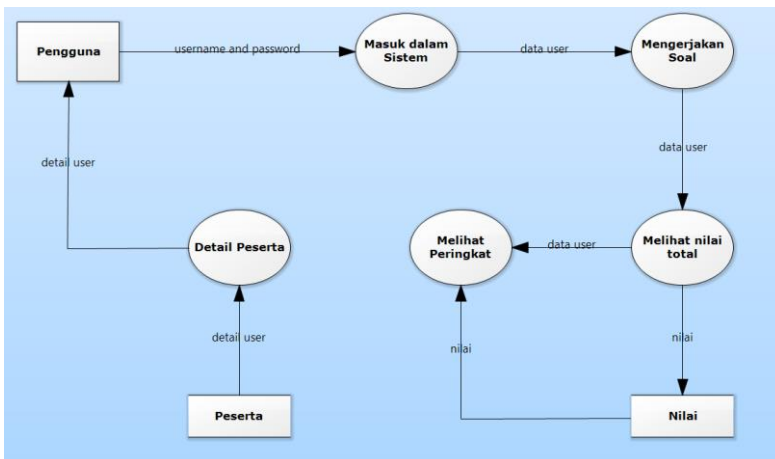
Dalam DFD level 1, secara keseluruhan terbagi menjadi dua bagian besar, yaitu ketika pengguna masuk atau *login* dan ketika pengguna melakukan registrasi. Setelah melakukan proses registrasi, pengguna dapat masuk dan mengakses aplikasi untuk mengerjakan soal. DFD level 1 dapat dilihat pada Gambar 3.2 dan Gambar 3.3.



Gambar 3.1 DFD Level 0



Gambar 3.2 DFD Level 1 untuk Proses Registrasi



Gambar 3.3 DFD Level 1

### 3.1.3.4 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional yang harus dipenuhi oleh sistem sebagai berikut:

1. Kebutuhan Performa  
Perangkat lunak akan berjalan dengan performa terbaik jika dijalankan di atas spesifikasi minimal.
2. Kebutuhan Perlindungan Keamanan  
*Username* dan *password* dibutuhkan untuk mengamankan data pengguna yang berhak mengakses sistem sebagai syarat memasuki sistem dan melakukan semua fungsionalitas sistem.
3. Kualitas perangkat lunak dapat dilihat pada Tabel 3.2.

**Tabel 3.2 Kualitas Perangkat Lunak**

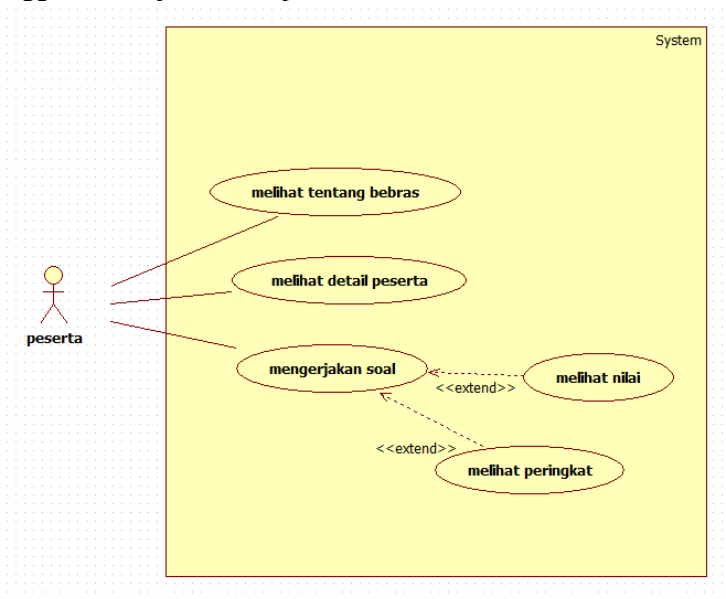
No	Parameter	Deskripsi
1	Kegunaan	Pada aspek ini indikator yang digunakan adalah kemudahan penggunaan menu, kemudahan dalam mendapatkan informasi, dan kesan <i>immersive</i>
2	Tingkat kualitas	Aplikasi dibangun dengan antarmuka pengguna yang konsisten, mudah dipahami dan mudah dioperasikan
3	Bahasa penyampaian	Terdapat dua bahasa yang tersedia ketika mengerjakan soal yaitu bahasa Indonesia dan bahasa Inggris, peserta dapat memilih salah satu bahasa yang telah tersedia tersebut.

### 3.1.3.5 Aktor

Pengertian pengguna adalah pihak-pihak yang terlibat dan berinteraksi secara langsung dengan sistem. Pengguna pada aplikasi ini adalah peserta. Peserta merupakan seseorang yang menjalankan aplikasi ini.

### 3.1.3.6 Kasus Penggunaan

Kasus penggunaan sistem dibuat berdasarkan analisis spesifikasi kebutuhan fungsional dan analisis aktor dari sistem. Kasus penggunaan digambarkan dalam tabel penjelasan kasus penggunaan dan diagram kasus penggunaan. Tabel penjelasan kasus penggunaan dapat dilihat pada Tabel 3.3 dan diagram kasus penggunaan dapat dilihat pada Gambar 3.4.



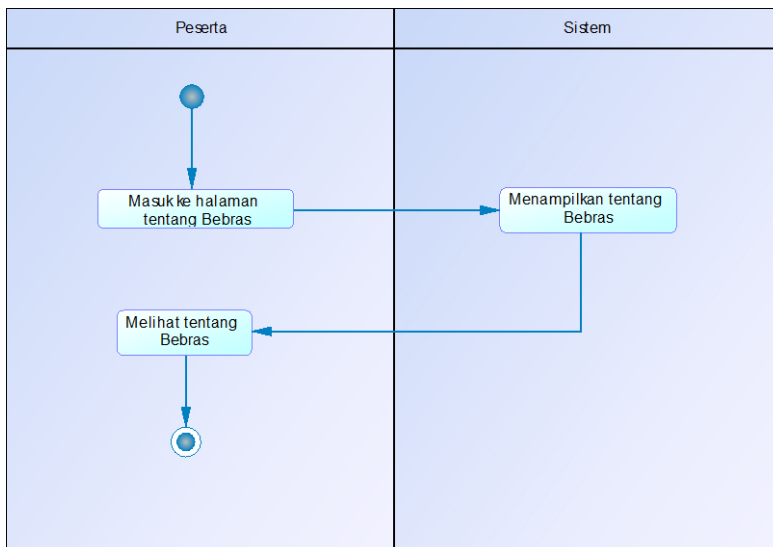
**Gambar 3.4 Diagram Kasus Penggunaan**

**Tabel 3.3 Kasus Penggunaan**

Kode Kasus Penggunaan	Nama	Aktor
UC-0001	Melihat tentang Bebras	Peserta
UC-0002	Menambah detail peserta	Peserta
UC-0003	Mengerjakan soal	Peserta
UC-0004	Melihat nilai	Peserta
UC-0005	Melihat peringkat	Peserta

### 3.1.3.6.1 Melihat Tentang Bebras (UC-0001)

Kasus penggunaan nomor UC-0001 diakses ketika peserta hendak melihat informasi tentang Bebras pada sistem. Spesifikasi dan diagram aktivitas kasus penggunaan ini dapat dilihat pada Tabel 3.4 dan Gambar 3.5.

**Gambar 3.5 Diagram Aktivitas Melihat Tentang Bebras**

**Tabel 3.4 Spesifikasi Kasus Penggunaan UC-0001**

<b>Kode Use Case</b>	UC-0001	
<b>Nama Use Case</b>	Melihat tentang Bebras	
<b>Aktor</b>	Peserta	
<b>Deskripsi</b>	Peserta dapat melihat tentang Bebras	
<b>Relasi</b>	-	
<b>Kondisi Awal</b>	Sistem belum menampilkan informasi tentang Bebras	
<b>Kondisi Akhir</b>	Sistem sudah menampilkan informasi tentang Bebras	
<b>Alur kejadian normal</b>	<b>Aktor</b>	<b>Sistem</b>
	1. Memilih pilihan “Tentang Bebras”	
		2. Menampilkan halaman yang berisi informasi tentang Bebras
<b>Alur kejadian alternatif</b>	<b>Aktor</b>	<b>Sistem</b>
	-	-

### 3.1.3.6.2 Melihat Detail Peserta (UC-0002)

Kasus penggunaan nomor UC-0002 diakses ketika peserta hendak melihat data peserta pada sistem. Spesifikasi, diagram aktivitas, *sequence* dan *robustness* kasus penggunaan ini dapat dilihat pada Tabel 3.5, Gambar 3.6, Gambar 3.7, dan Gambar 3.8.

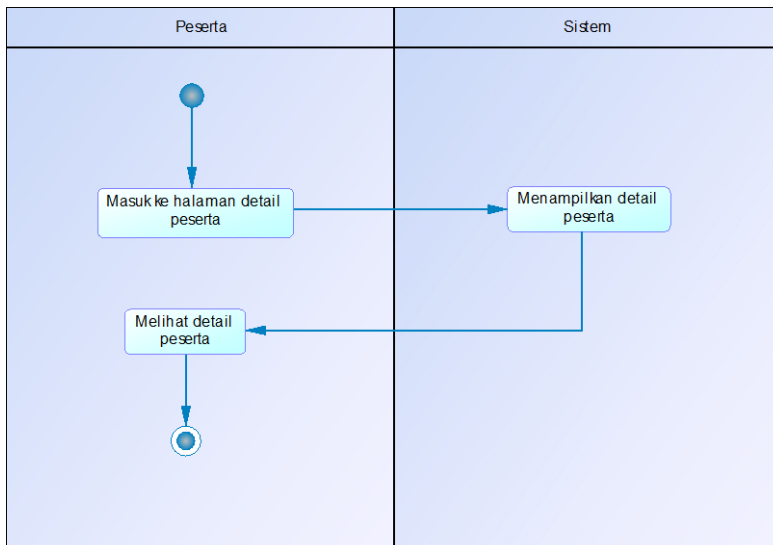
### 3.1.3.6.3 Mengerjakan Soal (UC-0003)

Kasus penggunaan nomor UC-0003 ini diakses ketika peserta hendak mengerjakan soal yang terdapat pada sistem. Spesifikasi, diagram aktivitas, *sequence*, dan *robustness* kasus penggunaan ini dapat dilihat pada Tabel 3.6 dan Tabel 3.7, Gambar 3.9, Gambar 3.10, dan Gambar 3.11.

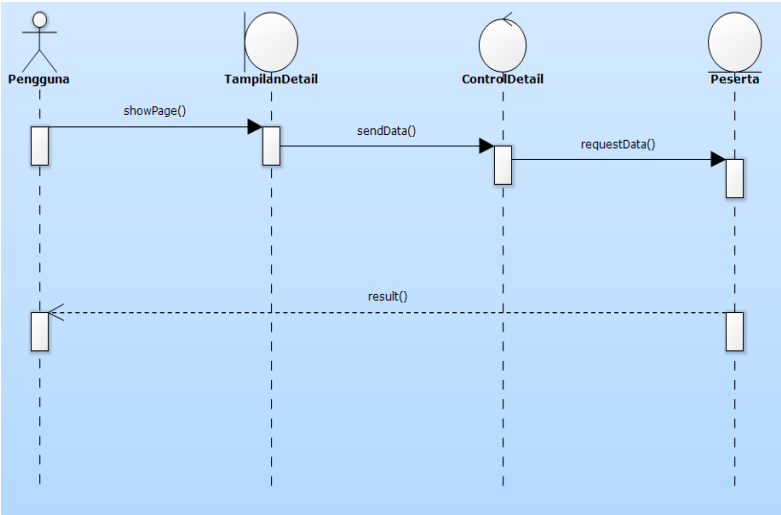


Tabel 3.5 Spesifikasi Kasus Penggunaan UC-0002

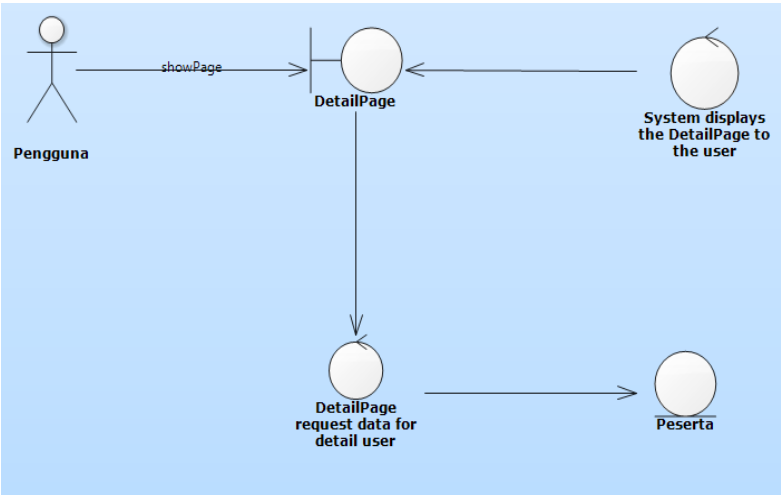
<b>Kode Use Case</b>	UC-0002	
<b>Nama Use Case</b>	Melihat detail peserta	
<b>Aktor</b>	Peserta	
<b>Deskripsi</b>	Peserta dapat melihat data peserta	
<b>Relasi</b>	-	
<b>Kondisi Awal</b>	Sistem belum menampilkan data peserta	
<b>Kondisi Akhir</b>	Sistem sudah menampilkan data peserta	
<b>Alur kejadian normal</b>	<b>Aktor</b>	<b>Sistem</b>
	1. Memilih pilihan “Detail Peserta”	
		2. Menampilkan halaman yang berisi data peserta
<b>Alur kejadian alternatif</b>	<b>Aktor</b>	<b>Sistem</b>
	-	-



Gambar 3.6 Diagram Aktivitas Melihat Detail Peserta



**Gambar 3.7 Diagram *Sequence* Melihat Detail Peserta**



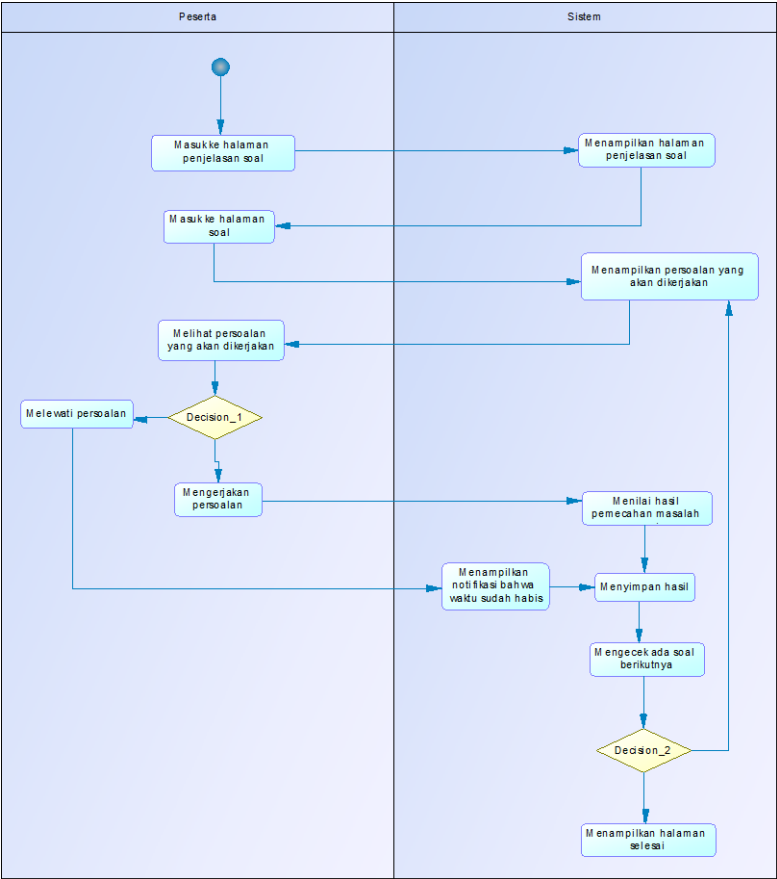
**Gambar 3.8 Diagram *Robustness* Melihat Detail Peserta**

Tabel 3.6 Spesifikasi Kasus Penggunaan UC-0003 (1)

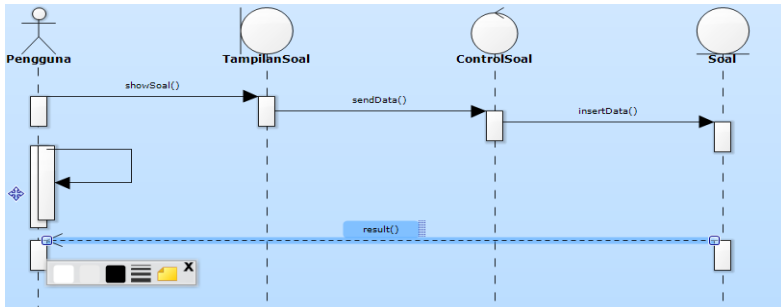
<b>Kode Use Case</b>	UC-0003	
<b>Nama Use Case</b>	Mengerjakan Soal	
<b>Aktor</b>	Peserta	
<b>Deskripsi</b>	Peserta dapat mengerjakan soal yang terdapat pada sistem	
<b>Relasi</b>	-	
<b>Kondisi Awal</b>	Peserta sudah melakukan <i>login</i>	
<b>Kondisi Akhir</b>	Peserta dapat mengerjakan soal	
<b>Alur kejadian normal</b>	<b>Aktor</b>	<b>Sistem</b>
	1.Masuk ke halaman Penjelasan Soal	
		2.Menampilkan halaman yang berisi tentang penjelasan soal
	3.Masuk ke halaman Soal	
		4.Menampilkan halaman yang berisi soal serta pilihan jawaban
	5.a.Mengerjakan persoalan	
		6. Menilai hasil pemecahan masalah
		7.Menyimpan hasil ke dalam <i>database</i>
		8.Mengecek ada soal berikutnya
		9.a.Menampilkan halaman Selesai
<b>Alur kejadian alternatif</b>	<b>Aktor</b>	<b>Sistem</b>
	5.b.Melewati persoalan	

Tabel 3.7 Spesifikasi Kasus Penggunaan UC-0003 (2)

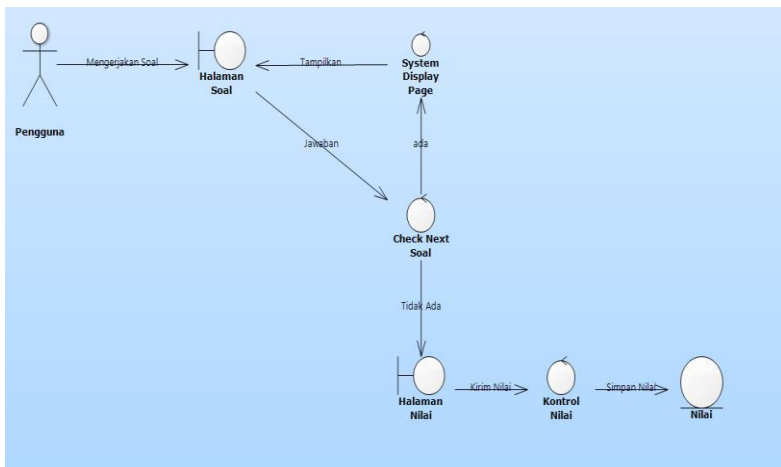
		5.b.1.Menampilkan notifikasi bahwa waktu habis
		5.b.2.Menuju alur 7
		9.b.Kembali ke alur 4



Gambar 3.9 Diagram Aktivitas Mengerjakan Soal



**Gambar 3.10 Diagram *Sequence* Mengerjakan Soal**



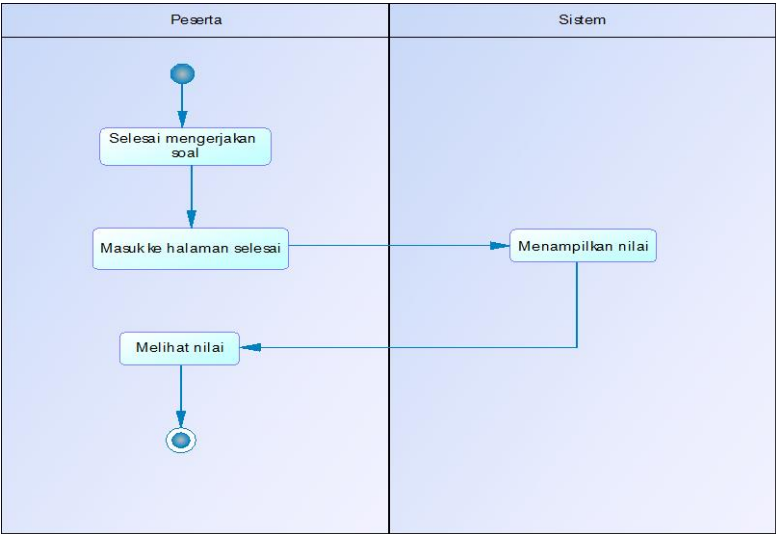
**Gambar 3.11 Diagram *Robustness* Mengerjakan Soal**

#### 3.1.3.6.4 Melihat Nilai (UC-0004)

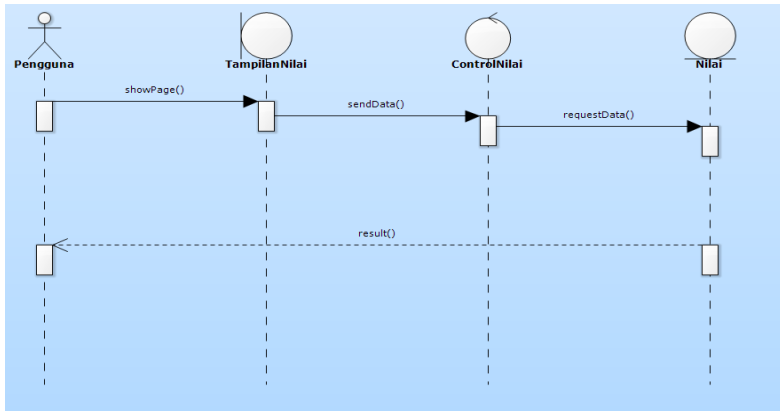
Kasus penggunaan nomor UC-0004 ini diakses ketika peserta ingin melihat nilai pada sistem. Spesifikasi, diagram aktivitas, *sequence* dan *robustness* kasus penggunaan ini dapat dilihat pada Tabel 3.8, Gambar 3.12, Gambar 3.13, dan Gambar 3.14.

Tabel 3.8 Spesifikasi Kasus Penggunaan UC-0004

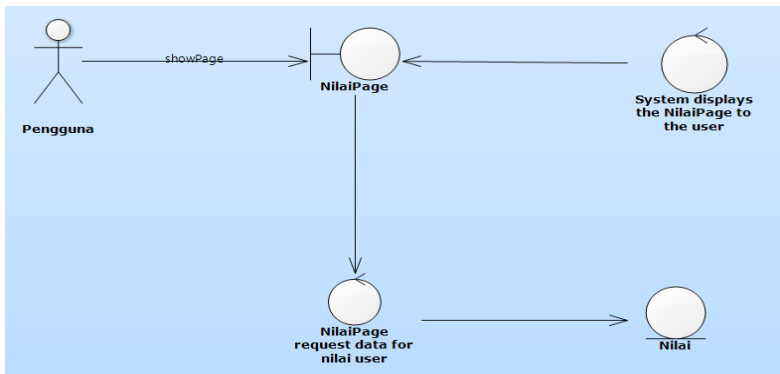
Kode Use Case	UC-0004	
Nama Use Case	Melihat nilai	
Aktor	Peserta	
Deskripsi	Peserta dapat melihat nilai	
Relasi	-	
Kondisi Awal	Sistem belum dapat menampilkan data nilai	
Kondisi Akhir	Sistem sudah dapat menampilkan data nilai	
Alur kejadian normal	Aktor	Sistem
	1. Selesai mengerjakan soal	
	2.Masuk ke halaman Selesai	
		3. Menampilkan halaman yang berisi nilai
Alur kejadian alternatif	Aktor	Sistem



Gambar 3.12 Diagram Aktivitas Melihat Nilai



**Gambar 3.13 Diagram *Sequence* Melihat Nilai**



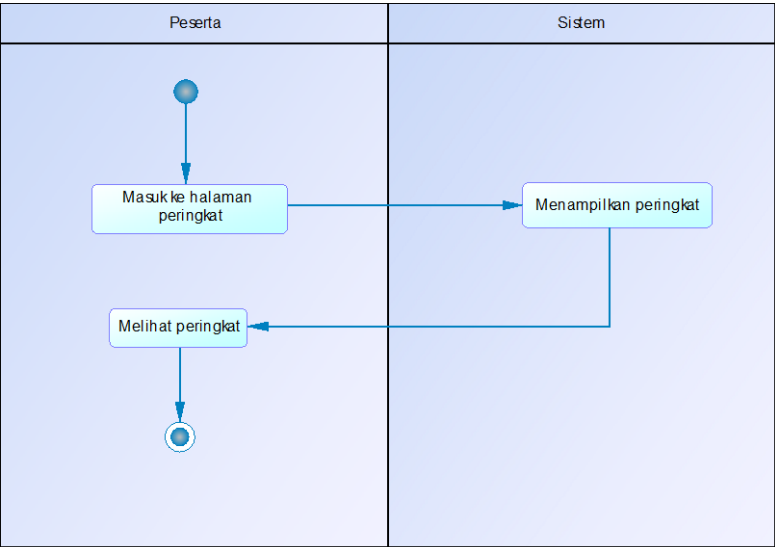
**Gambar 3.14 Diagram *Robustness* Melihat Nilai**

### 3.1.3.6.5 Melihat Peringkat (UC-0005)

Kasus penggunaan nomor UC-0005 ini diakses ketika peserta hendak melihat peringkat pada sistem. Spesifikasi, diagram aktivitas, *sequence* dan *robustness* kasus penggunaan ini dilihat pada Tabel 3.9, Gambar 3.15, Gambar 3.16, dan Gambar 3.17.

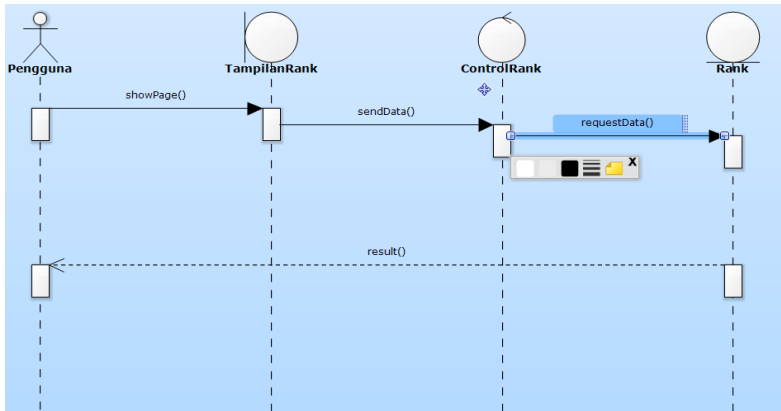
Tabel 3.9 Spesifikasi Kasus Penggunaan UC-0005

Kode Use Case	UC-0005	
Nama Use Case	Melihat peringkat	
Aktor	Peserta	
Deskripsi	Peserta dapat melihat peringkat	
Relasi	-	
Kondisi Awal	Sistem belum menampilkan data peringkat	
Kondisi Akhir	Sistem sudah menampilkan data peringkat	
Alur kejadian normal	Aktor	Sistem
	1.Memilih pilihan “RANKING”	
		2. Menampilkan halaman yang berisi data peringkat
Alur kejadian alternatif	Aktor	Sistem
	-	-

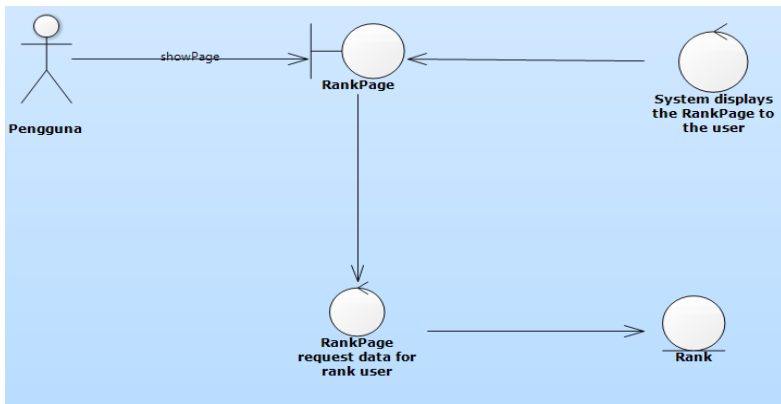


Gambar 3.15 Diagram Aktivitas Melihat Peringkat





**Gambar 3.16 Diagram *Sequence* Melihat Peringkat**



**Gambar 3.17 Diagram *Robustness* Melihat Peringkat**

### 3.2 Perancangan

Perancangan dalam subbab ini membahas perancangan dari aplikasi tugas akhir. Subbab ini terdiri dari lingkungan perancangan perangkat lunak, perancangan arsitektur sistem, perancangan basis data, dan perancangan antarmuka pengguna.

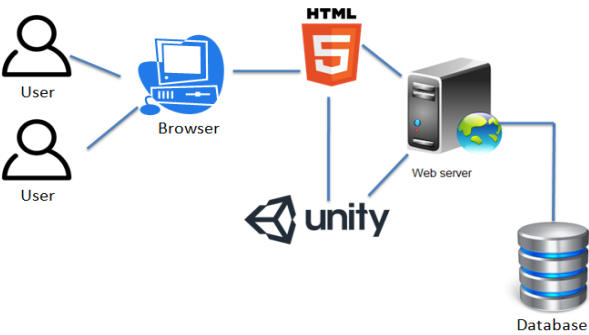
3.2.1 Lingkungan Perancangan Perangkat Lunak

Spesifikasi perangkat keras serta perangkat lunak yang digunakan dalam tahap perancangan perangkat lunak tugas akhir ini seperti dijelaskan pada Tabel 3.10.

Tabel 3.10 Lingkungan Perancangan Perangkat Lunak

Perangkat Keras	1. Laptop Vaio 2. Processor Intel(R) Core(TM) i7-2640M CPU @ 2.80GHz 3. Memori (RAM) 4.00 GB
Perangkat Lunak	1. System operasi Windows 7 (64-bit) 2. Unity 5.5.2 3. Visual Studio 2015 4. Google Chrome Version 59.0.3071.115 5. Mozilla Firefox Version 54.0.1 6. Sybase PowerDesigner Version 16.5 7. Microsoft Word 2010 8. Sublime 9. StarUML version 5.0.2.1570

3.2.2 Perancangan Arsitektur Sistem



Gambar 3.18 Arsitektur Sistem

Pada arsitektur sistem ini, untuk mengakses aplikasi ini membutuhkan *browser* seperti Firefox dan Chrome dan terhubung ke internet. Perancangan arsitektur sistem pada aplikasi ini dapat dilihat pada Gambar 3.18. Soal yang dibuat menggunakan *game engine* Unity kemudian *build* ke dalam HTML5 agar bisa dibuka dalam bentuk *web*. Kemudian melalui *web service* berupa PHP dilakukan penyimpanan ke dalam *database*.

Dalam proses aktivitas aplikasi, aplikasi dapat dibuka dalam bentuk *web*, kemudian data akan disimpan melalui *web service* ke dalam *database*.

### 3.2.3 Perancangan Basis Data

Dalam membuat suatu aplikasi perangkat bergerak, diperlukan analisis kebutuhan berupa perancangan basis data. Basis data yang digunakan adalah MySQL. Rancangan basis data ditampilkan dalam bentuk *Conceptual Data Model* (CDM) dan *Physical Data Model* (PDM). Penjelasan lebih lengkap berupa CDM dan PDM terdapat pada Gambar 3.19 dan Gambar 3.20.

#### 3.2.3.1 Rancangan Tabel Peserta

Tabel peserta adalah tabel yang digunakan untuk menyimpan data-data peserta. Tabel peserta mempunyai relasi ke tabel jawaban. Satu peserta yang terdaftar pada sistem memiliki banyak jawaban. Data atribut pada tabel peserta dijelaskan di Tabel 3.11.

**Tabel 3.11 Atribut Tabel Peserta**

<b>Nama Kolom</b>	<b>Tipe Data</b>	<b>Keterangan</b>
id	Integer	<i>Primary Key</i> dari tabel peserta
username	Varchar(1024)	Nama akun peserta
nama	Varchar(1024)	Nama peserta
provinsi	Varchar(1024)	Provinsi peserta
nilai	Integer	Total nilai yang diperoleh peserta

### 3.2.3.2 Rancangan Tabel Soal

Tabel soal adalah tabel yang digunakan untuk menyimpan data-data soal. Tabel soal mempunyai relasi ke tabel jawaban. Satu soal yang terdapat pada sistem memiliki banyak jawaban. Data atribut pada tabel soal dijelaskan pada Tabel 3.12.

**Tabel 3.12 Atribut Tabel Soal**

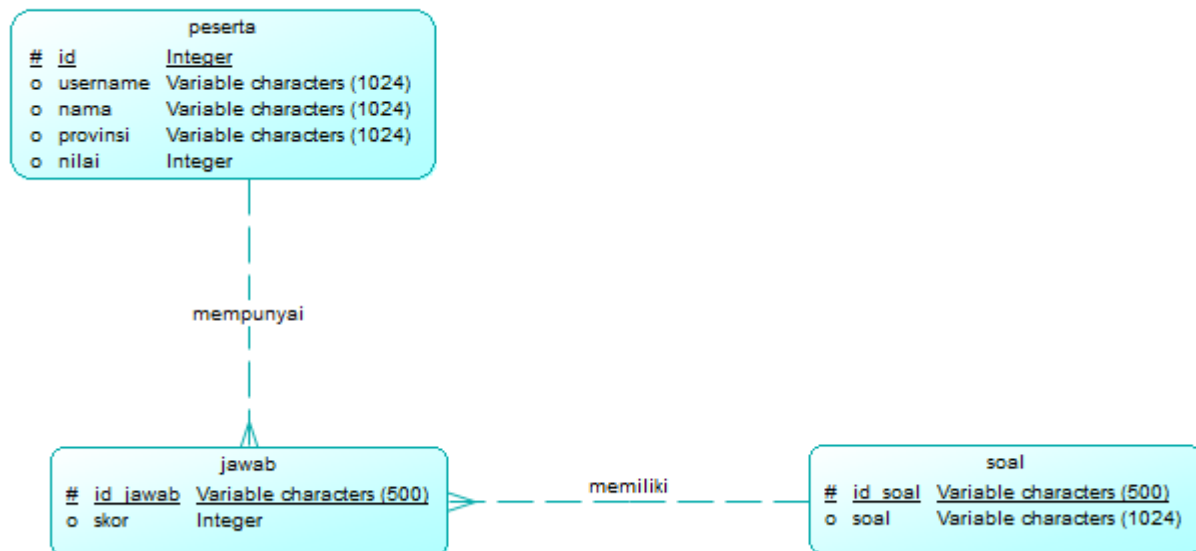
<b>Nama Kolom</b>	<b>Tipe Data</b>	<b>Keterangan</b>
id_soal	Varchar(500)	<i>Primary Key</i> dari tabel soal
soal	Varchar(1024)	Nama soal

### 3.2.3.3 Rancangan Tabel Jawab

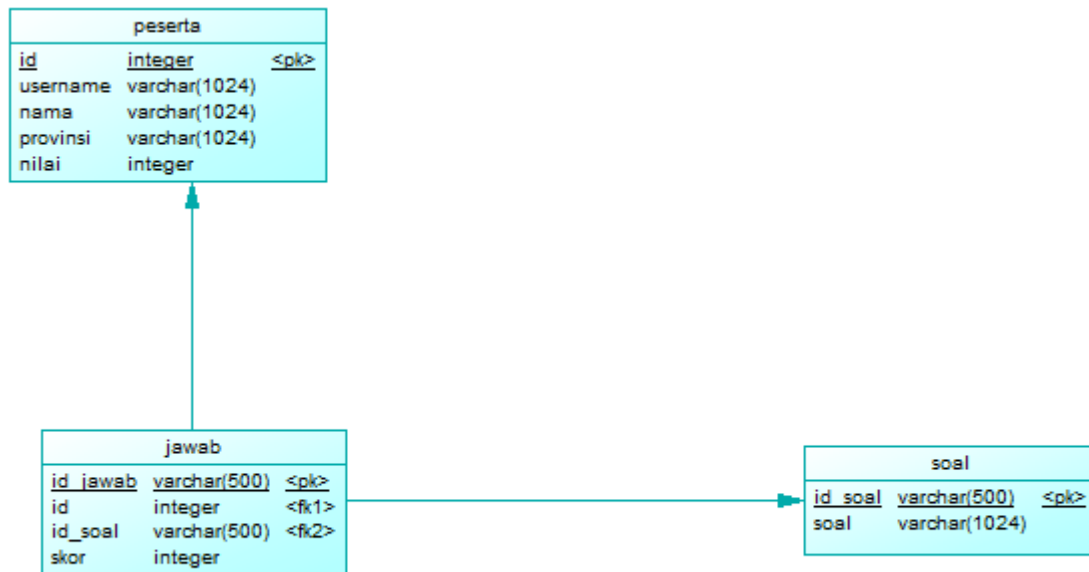
Tabel jawab adalah tabel yang digunakan untuk menyimpan nilai dari jawaban peserta. Tabel jawab mempunyai relasi ke tabel peserta dan tabel soal. Satu peserta memiliki banyak jawaban serta satu soal memiliki banyak jawaban. Data atribut pada tabel soal dijelaskan pada Tabel 3.13.

**Tabel 3.13 Atribut Tabel Jawab**

<b>Nama Kolom</b>	<b>Tipe Data</b>	<b>Keterangan</b>
id_jawab	Varchar(500)	<i>Primary Key</i> dari tabel jawab
id	Integer	<i>Foreign Key</i> dari tabel peserta
id_soal	Varchar(500)	<i>Foreign Key</i> dari tabel soal
skor	Integer	Nilai yang diperoleh peserta



Gambar 3.19 *Conceptual Data Model*



**Gambar 3.20** *Physical Data Model*

### 3.2.4 Perancangan Antarmuka Pengguna

Perancangan antarmuka pengguna merupakan hal yang penting dalam melakukan perancangan perangkat lunak. Antarmuka pengguna yang berhubungan langsung dengan aktor harus memiliki kemudahan-kemudahan dan tampilan yang menarik bagi penggunanya. Berdasarkan pengguna yang menggunakan aplikasi, maka terdapat antarmuka untuk peserta.

#### 3.2.4.1 Rancangan Antarmuka Halaman Register Pengguna

Halaman ini digunakan oleh pengguna yaitu peserta mendaftarkan akun baru untuk bisa melakukan pengerjaan soal. Peserta harus mengisi isian *username*, nama, *password*, ulangi *password*, dan memilih *dropdown* provinsi untuk dapat masuk ke dalam sistem. Rancangan antarmuka halaman Register ditunjukkan pada Gambar 3.21. Penjelasan pada masing-masing nomor yang tertera pada Gambar 3.21 dijelaskan pada atribut antarmuka Register yang dapat dilihat pada Tabel 3.14.

**Tabel 3.14 Atribut Antarmuka Halaman Register Pengguna**

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan
1	<i>Username</i>	<i>InputField</i>	<i>Input</i> untuk memasukkan <i>username</i> pengguna
2	Nama	<i>InputField</i>	<i>Input</i> untuk memasukkan nama pengguna
3	<i>Password</i>	<i>InputField</i>	<i>Input</i> untuk memasukkan <i>password</i>
4	<i>RePassword</i>	<i>InputField</i>	<i>Input</i> untuk memasukkan ulangi <i>password</i>
5	Provinsi	<i>Dropdown</i>	<i>Input</i> untuk memilih provinsi
6	<i>btnRegister</i>	<i>Button</i>	Tombol aksi untuk menyimpan data ke <i>database</i>
7	<i>btnKembali</i>	<i>Button</i>	Tombol aksi untuk menuju halaman Login

The image shows a registration form titled "REGISTER" on a light purple background. It contains the following elements with numbered callouts:

- 1: Username input field
- 2: Nama (Name) input field
- 3: Password input field
- 4: Ulangi Password (Repeat Password) input field
- 5: Dropdown menu labeled "Pilih Provinsi Anda" (Select Your Province)
- 6: Green "Register" button
- 7: White "Kembali" (Back) button

**Gambar 3.21 Rancangan Antarmuka Halaman Register**

### 3.2.4.2 Rancangan Antarmuka Halaman Login Pengguna

Halaman ini digunakan oleh pengguna untuk bisa melakukan pengerjaan soal. Pengguna harus mengisi isian *username* dan *password* untuk dapat masuk ke dalam sistem. Rancangan antarmuka halaman Login ditunjukkan pada Gambar 3.22. Penjelasan pada masing-masing nomor yang tertera pada Gambar 3.22 dijelaskan pada atribut antarmuka Login yang dapat dilihat pada Tabel 3.15.

The image shows a login form titled "LOGIN" on a light purple background. It contains the following elements with numbered callouts:

- 1: Username input field
- 2: Password input field
- 3: Blue "Login" button
- 4: Text link: "Belum punya akun? klik tombol Register"
- 5: Green "Register" button
- 6: White "Home" button

**Gambar 3.22 Rancangan Antarmuka Halaman Login**



**Tabel 3.15 Atribut Antarmuka Halaman Login Pengguna**

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan
1	<i>Username</i>	<i>InputField</i>	<i>Input</i> untuk memasukkan <i>username</i> pengguna
2	<i>Password</i>	<i>InputField</i>	<i>Input</i> untuk memasukkan <i>password</i> pengguna
3	<i>btnLogin</i>	<i>Button</i>	Tombol aksi untuk <i>login</i> menuju ke halaman Penjelasan Soal
4	<i>text</i>	<i>Text</i>	Tulisan 'Belum punya akun? Klik tombol register'
5	<i>btnRegis</i>	<i>Button</i>	Tombol aksi untuk menuju ke halaman Register
6	<i>btnKembaliHome</i>	<i>Button</i>	Tombol aksi untuk menuju ke halaman Beranda

### 3.2.4.3 Rancangan Antarmuka Halaman Beranda

Halaman ini akan ditampilkan pada sistem saat pengguna pertama kali membuka *web*. Rancangan antarmuka halaman Beranda ditunjukkan pada Gambar 3.23. Penjelasan pada masing-masing nomor yang tertera pada Gambar 3.23 dijelaskan pada atribut antarmuka Beranda yang dapat dilihat pada Tabel 3.16.

**Tabel 3.16 Atribut Antarmuka Halaman Beranda**

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan
1	<i>btnHome</i>	<i>Button</i>	Tombol aksi untuk menuju ke halaman Beranda
2	<i>btnKompetisi</i>	<i>Button</i>	Tombol aksi untuk menuju ke halaman Kompetisi
3	<i>btnInfo</i>	<i>Button</i>	Tombol aksi untuk menuju ke halaman Tentang Bebras
4	<i>btnDetailPeserta</i>	<i>Button</i>	Tombol aksi untuk menuju ke halaman Detail Peserta



Gambar 3.23 Rancangan Antarmuka Halaman Beranda

3.2.4.4 Rancangan Antarmuka Halaman Kompetisi

Halaman ini akan ditampilkan untuk masuk kompetisi melalui *login* terlebih dahulu. Rancangan antarmuka halaman Kompetisi ditunjukkan pada Gambar 3.24. Penjelasan pada masing-masing nomor yang tertera pada Gambar 3.24 dijelaskan pada atribut antarmuka Kompetisi yang dapat dilihat pada Tabel 3.17.

Tabel 3.17 Atribut Antarmuka Halaman Kompetisi

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan
1	<i>btnHome</i>	<i>Button</i>	Tombol aksi untuk menuju ke halaman Beranda
2	<i>btnKompetisi</i>	<i>Button</i>	Tombol aksi untuk menuju ke halaman Kompetisi
3	<i>btnInfo</i>	<i>Button</i>	Tombol aksi untuk menuju ke halaman Tentang Bebras
4	<i>btnDetailPeserta</i>	<i>Button</i>	Tombol aksi untuk menuju ke halaman Detail Peserta



**Gambar 3.24 Rancangan Antarmuka Halaman Kompetisi**

### 3.2.4.5 Rancangan Antarmuka Halaman Detail Peserta

Pada halaman ini akan ditampilkan data peserta, jumlah peserta dan jumlah per provinsi. Rancangan antarmuka halaman Detail Peserta ditunjukkan pada Gambar 3.25. Penjelasan pada masing-masing nomor yang tertera pada Gambar 3.25 dijelaskan pada atribut antarmuka Detail Peserta yang dapat dilihat pada Tabel 3.18.

**Tabel 3.18 Atribut Antarmuka Halaman Detail Peserta**

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan
1	<i>jmlpeserta</i>	<i>Text</i>	Informasi jumlah peserta
2	<i>Usernamepeserta</i>	<i>Text</i>	Informasi <i>username</i> , nama, provinsi dari peserta
3	<i>prov</i>	<i>Text</i>	Informasi jumlah per provinsi
4	<i>btnKembali</i>	<i>Button</i>	Tombol aksi untuk menuju ke halaman Beranda



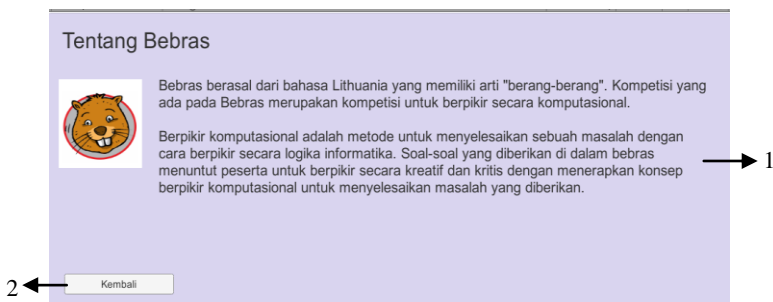
**Gambar 3.25 Rancangan Antarmuka Halaman Detail Peserta**

### 3.2.4.6 Rancangan Antarmuka Halaman Tentang Bebras

Pada halaman ini akan ditampilkan informasi tentang Bebras. Rancangan antarmuka halaman Tentang Bebras ditunjukkan pada Gambar 3.26. Penjelasan pada masing-masing nomor yang tertera pada Gambar 3.26 dijelaskan pada atribut antarmuka Tentang Bebras yang dapat dilihat pada Tabel 3.19.

**Tabel 3.19 Atribut Antarmuka Halaman Tentang Bebras**

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan
1	<i>Text</i>	<i>Text</i>	Informasi mengenai Bebras
2	<i>btnKembali</i>	<i>Button</i>	Tombol aksi untuk menuju halaman Beranda



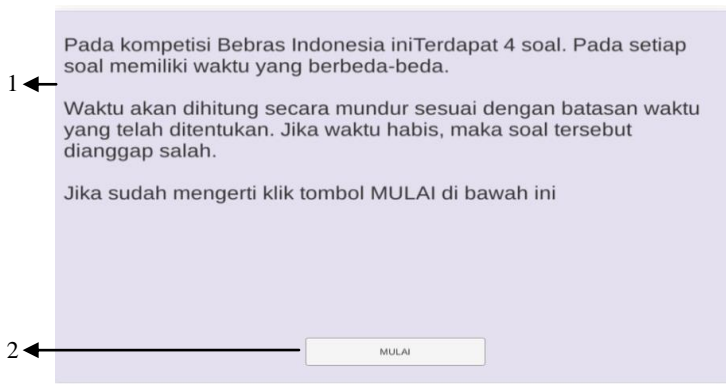
**Gambar 3.26 Rancangan Antarmuka Halaman Tentang Bebras**

### 3.2.4.7 Rancangan Antarmuka Halaman Penjelasan Soal

Pada halaman ini akan ditampilkan informasi tentang penjelasan soal. Rancangan antarmuka halaman Penjelasan Soal ditunjukkan pada Gambar 3.27. Penjelasan pada masing-masing nomor yang tertera pada Gambar 3.27 dijelaskan pada atribut antarmuka Penjelasan Soal yang dapat dilihat pada Tabel 3.20.

**Tabel 3.20 Atribut Antarmuka Halaman Penjelasan Soal**

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan
1	<i>Text</i>	<i>Text</i>	Informasi tentang penjelasan soal
2	<i>btnno1</i>	<i>Button</i>	Tombol aksi untuk menuju halaman Soal



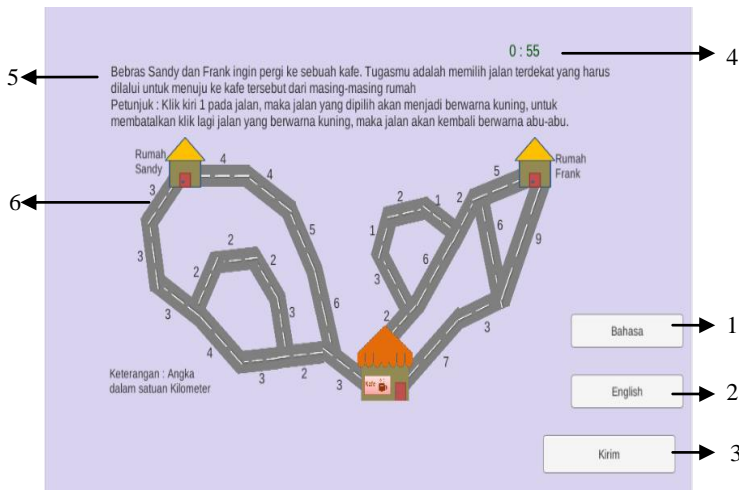
**Gambar 3.27 Rancangan Antarmuka Halaman Penjelasan Soal**

### 3.2.4.8 Rancangan Antarmuka Halaman Soal

Pada halaman ini akan ditampilkan salah satu contoh uraian soal dan pilihan jawaban. Rancangan antarmuka halaman Soal ditunjukkan pada Gambar 3.28. Penjelasan pada masing-masing nomor yang tertera pada Gambar 3.28 dijelaskan pada atribut antarmuka Soal yang dapat dilihat pada Tabel 3.21.

Tabel 3.21 Atribut Antarmuka Halaman Soal

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan
1	<i>btnBahasa</i>	<i>Button</i>	Tombol aksi untuk menerjemahkan soal dalam bahasa Indonesia
2	<i>btnInggris</i>	<i>Button</i>	Tombol aksi untuk menerjemahkan soal dalam bahasa Inggris
3	<i>btnSubmit</i>	<i>Button</i>	Tombol aksi untuk menuju halaman Soal berikutnya
4	<i>txtTimer</i>	<i>Text</i>	Teks untuk hitung mundur waktu soal
5	<i>text</i>	<i>Text</i>	Teks soal
6	<i>btnno1</i>	<i>Button</i>	Tombol untuk memilih jalan



Gambar 3.28 Rancangan Antarmuka Halaman Soal

### 3.2.4.9 Rancangan Antarmuka Halaman Selesai

Pada halaman ini akan ditampilkan nilai yang didapatkan oleh pengguna. Rancangan antarmuka halaman Selesai ditunjukkan pada Gambar 3.29. Penjelasan pada masing-masing nomor yang tertera pada Gambar 3.29 dijelaskan pada atribut antarmuka Selesai yang dapat dilihat pada Tabel 3.22.

**Tabel 3.22 Atribut Antarmuka Halaman Selesai**

<i>No.</i>	<i>Nama Atribut Antarmuka</i>	<i>Jenis Atribut</i>	<i>Kegunaan</i>
1	Nilai	Text	Informasi total nilai
2	Btnrating	Button	Tombol aksi menuju halaman Peringkat
3	btnLogout	Button	Tombol aksi keluar dari sistem pengerjaan soal dan menuju halaman Beranda



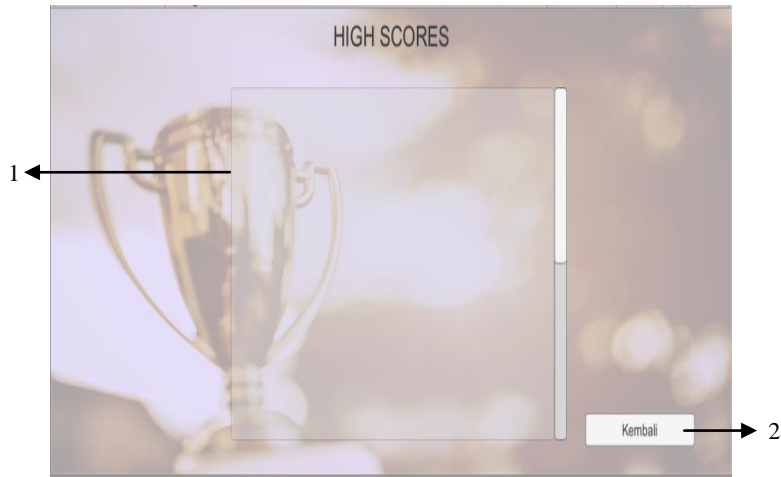
**Gambar 3.29 Rancangan Antarmuka Halaman Selesai**

3.2.4.10 Rancangan Antarmuka Halaman Peringkat

Pada halaman ini akan ditampilkan peringkat berupa nilai dari semua peserta. Rancangan antarmuka halaman Peringkat ditunjukkan pada Gambar 3.30. Penjelasan pada masing-masing nomor yang tertera pada Gambar 3.30 dijelaskan pada atribut antarmuka Peringkat yang dapat dilihat pada Tabel 3.23.

Tabel 3.23 Atribut Antarmuka Halaman Peringkat

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan
1	<i>Text</i>	<i>Text</i>	Informasi peringkat peserta berupa semua nilai peserta
2	<i>btnKembali</i>	<i>Button</i>	Tombol aksi menuju halaman Selesai



Gambar 3.30 Rancangan Antarmuka Halaman Peringkat



## **BAB IV IMPLEMENTASI**

Bab ini membahas implementasi yang dilakukan berdasarkan rancangan yang telah dijabarkan pada bab sebelumnya. Sebelum penjelasan implementasi akan ditunjukkan terlebih dahulu lingkungan untuk melakukan implementasi. Pada bagian implementasi ini juga akan dijelaskan mengenai fungsi-fungsi yang digunakan dalam program tugas akhir ini dan disertai dengan kode sumber masing-masing fungsi utama.

### **4.1 Lingkungan Implementasi Perangkat Lunak**

Spesifikasi perangkat keras serta perangkat lunak yang digunakan dalam tahap implementasi perangkat lunak tugas akhir ini seperti dijelaskan pada Tabel 4.1.

**Tabel 4.1 Lingkungan Implementasi Perangkat Lunak**

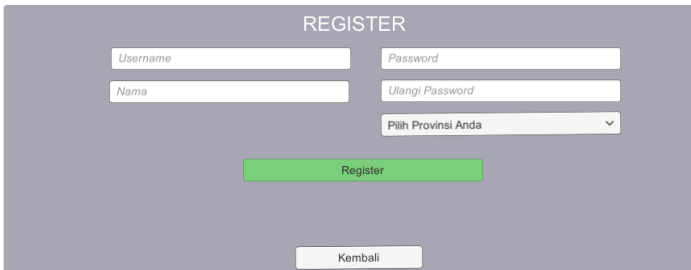
Perangkat Keras	<ol style="list-style-type: none"><li>1. Laptop Vaio</li><li>2. Processor Intel(R) Core(TM) i7-2640M CPU @ 2.80GHz</li><li>3. Memori (RAM) 4.00 GB</li></ol>
Perangkat Lunak	<ol style="list-style-type: none"><li>1. System operasi Windows 7 (64-bit)</li><li>2. Unity 5.5.2</li><li>3. Visual Studio 2015</li><li>4. Google Chrome Version 59.0.3071.115</li><li>5. Mozilla Firefox Version 54.0.1</li><li>6. Sybase PowerDesigner Version 16.5</li><li>7. Microsoft Word 2010</li><li>8. Sublime</li><li>9. StarUML version 5.0.2.1570</li></ol>

## 4.2 Implementasi Antarmuka Pengguna

Implementasi antarmuka pengguna berbasis *web* ini menggunakan HTML5. Subbab ini akan membahas tentang implementasi antarmuka. Antarmuka yang dibuat berdasarkan rancangan antarmuka yang telah dijabarkan pada Bab 3.

### 4.2.1 Implementasi Antarmuka Halaman Register Pengguna

Antarmuka halaman Register pada Gambar 4.1 merupakan halaman yang digunakan oleh pengguna untuk registrasi ke sistem. Terdapat tempat isian untuk *username*, nama, *password*, Ulangi *password*, *dropdown* provinsi, tombol Register serta tombol Kembali. Terdapat kode sumber fungsi register yang dapat dilihat pada Kode Sumber 4.1 dan kode sumber kontrol register pengguna pada Kode Sumber 4.2.



**Gambar 4.1 Implementasi Antarmuka Halaman Register Pengguna**

```

1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine.Networking;
4. using UnityEngine;
5. using UnityEngine.UI;
6. using System;
7.
8. public class Registrasi : MonoBehaviour {
9.

```

```

10.     List<string> names = new List<string>() { "Pili
      h Provinsi Anda", "ACEH", "SUMATERA UTARA", "SUMATE
      RA BARAT",
11.         "RIAU", "JAMBI", "SUMATERA SELATAN", "BENGK
      ULU", "LAMPUNG", "KEPULAUAN BANGKA BELITUNG", "KEPU
      LAUAN RIAU", "DKI JAKARTA",
12.         "JAWA BARAT", "JAWA TENGAH", "DI YOGYAKARTA
      ", "JAWA TIMUR", "BANTEN", "BALI", "NUSA TENGGARA B
      ARAT", "NUSA TENGGARA TIMUR",
13.         "KALIMANTAN BARAT", "KALIMANTAN TENGAH", "K
      ALIMANTAN SELATAN", "KALIMANTAN TIMUR", "KALIMANTAN
      UTARA", "SULAWESI UTARA",
14.         "SULAWESI TENGAH", "SULAWESI SELATAN", "SUL
      AWESI TENGGARA", "GORONTALO", "SULAWESI BARAT", "MA
      LUKU", "MALUKU UTARA",
15.         "PAPUA BARAT", "PAPUA"
16.     };
17.
18.     public Dropdown dropdown;
19.     public Text pilihprovinsi;
20.
21.     public InputField r_username;
22.     public InputField r_name;
23.     public InputField r_pass;
24.     public InputField r_repass;
25.
26.     private string username_r;
27.     private string name_r;
28.     private string pass_r;
29.     private string repass_r;
30.
31.     private string provinsi;
32.
33.     public Text error;
34.
35.     private string g_string;
36.     private Color green = Color.blue;
37.     private Color red = new Color(64, 0, 0, 1);
38.
39.     public void Dropdown_indexchanged(int index)
40.     {
41.         pilihprovinsi.text = names [index];
42.

```

```

43.     }
44.
45.
46.     void Start()
47.     {
48.         PopulateList ();
49.     }
50.
51.     void PopulateList(){
52.         dropdown.AddOptions (names);
53.     }
54.
55.     public void submit()
56.     {
57.         error.text = "";
58.         username_r = r_username.text;
59.         name_r = r_name.text;
60.         pass_r = r_pass.text;
61.         repass_r = r_repass.text;
62.         provinsi = pilihprovinsi.text;
63.
64.         //sign_up(username_r, name_r, pass_r);
65.         StartCoroutine(sign_up(username_r, name_r,
pass_r, provinsi, 5f));
66.     }
67.
68.     IEnumerator sign_up(string elm, string elm1, st
ring elm2, string elm3, float delay)
69.     {
70.         if (pass_r == repass_r)
71.         {
72.             if (provinsi != "Pilih Provinsi Anda")
73.             {
74.                 UnityWebRequest link = UnityWebRequ
est.Get("http://somethingnotright.dx.am/registrasi.
php?r_username=" + elm + "&r_nama=" + elm1 + "&r_pa
ss="+elm2 + "&r_provinsi="+elm3);
75.                 yield return link.Send();
76.                 yield return new WaitForSeconds(del
ay);
77.                 if (link.isError)
78.                 {

```

```

79.             Debug.Log(link.error);
80.         }
81.         else
82.         {
83.             g_string = link.downloadHandler
84.             .text;
85.             show_error(g_string.ToString())
86.         ;
87.         }
88.         else error.text = "Silahkan Pilih Provi
89.         nsi Anda";
90.         else error.text = "Password Tidak Cocok";
91.     }
92.
93.     private void show_error(string elm)
94.     {
95.         string[] arr = null;
96.         arr = elm.Split(new string[] { " " }, Strin
97.         gSplitOptions.RemoveEmptyEntries);
98.         if (arr [3] == "Berhasil") {
99.             error.color = green;
100.            error.text = elm;
101.        }
102.        else
103.        {
104.            error.color = red;
105.            error.text = elm;
106.        }
107.    }

```

#### Kode Sumber 4.1 Fungsi Register Pengguna

```

1. <?php
2.
3. header("Access-Control-Allow-Credentials: true");
4. header('Access-Control-Allow-Origin: *');

```

```

5. header('Access-Control-Allow-
   Methods: POST, GET, OPTIONS');
6. header('Access-Control-Allow-Headers: Accept, X-
   Access-Token, X-Application-Name, X-Request-Sent-
   Time');
7.
8. error_reporting(E_ALL ^ E_DEPRECATED);
9.
10. $Server = "fdb17.awardspace.net";
11. $User = "2326813_admin";
12. $Pass = "zzz1312000";
13. $Db = "2326813_admin";
14. $Connection = mysql_connect($Server, $User, $Pass,
   $Db);
15.
16. if(mysql_error()){
17.     die("Connection Failed!=". mysql_error());
18. }
19. mysql_select_db("2326813_admin", $Connection) or di
   e("Could not load to database" . mysql_error());
20.
21. $username = $_GET["r_username"];
22. $nama = $_GET["r_nama"];
23. $pass = $_GET["r_pass"];
24. $prov = $_GET["r_provinsi"];
25.
26. $check = mysql_query("SELECT * FROM peserta WHERE `
   username` = '". $username.'"' );
27. $numrows = mysql_num_rows($check);
28.
29. if((empty($username))||(empty($nama))||(e
   mpty($pass))||(empty($prov))){
30.     echo "Data yang Anda masukkan kurang";
31.     die;
32. }
33. elseif($numrows==0)
34. {
35.     $sql = mysql_query("INSERT INTO peserta (id, us
   ername, name, password, provinsi) VALUES ('', '". $u
   sername.'', '','$nama.', ' '.md5(md5($pass)).'', ' '.
   $prov.'")");

```

```

36.     $soal0 = mysql_query("INSERT INTO jawab (id_jaw
    ab, id, id_soal) VALUES (CONCAT(LAST_INSERT_ID(), 'S
    0'), LAST_INSERT_ID(), 'S0')");
37.     $soal1 = mysql_query("INSERT INTO jawab (id_jaw
    ab, id, id_soal) VALUES (CONCAT(LAST_INSERT_ID(), 'S
    1'), LAST_INSERT_ID(), 'S1')");
38.     $soal2 = mysql_query("INSERT INTO jawab (id_jaw
    ab, id, id_soal) VALUES (CONCAT(LAST_INSERT_ID(), 'S
    2'), LAST_INSERT_ID(), 'S2')");
39.     $soal3 = mysql_query("INSERT INTO jawab (id_jaw
    ab, id, id_soal) VALUES (CONCAT(LAST_INSERT_ID(), 'S
    3'), LAST_INSERT_ID(), 'S3')");
40.
41.
42.     if($sql){
43.         echo "memasukkan username ".$username." Ber
    hasil";
44.     }
45.     else{
46.         die("memasukkan username ".$username." Gaga
    1");
47.     }
48. }
49. else
50. {
51.     die("Username sudah ada !");
52. }
53.
54. mysql_close($Connection);
55.
56. ?>

```

#### Kode Sumber 4.2 Kontrol Register Pengguna

#### 4.2.2 Implementasi Antarmuka Halaman Login Pengguna

Antarmuka halaman Login pada Gambar 4.2 merupakan halaman yang digunakan oleh pengguna untuk masuk ke sistem dan mengakses halaman Soal. Terdapat tempat isian untuk *username*, *password*, tombol Login tombol Register serta tombol Kembali. Terdapat kode sumber fungsi login yang dapat dilihat

pada Kode Sumber 4.3 dan kode sumber kontrol login pengguna pada Kode Sumber 4.4.

**Gambar 4.2 Implementasi Antarmuka Halaman Login Pengguna**

```

1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using UnityEngine.UI;
5. using System;
6. using UnityEngine.Networking;
7.
8.
9. public class Login : MonoBehaviour {
10.
11.     public Button submitBtn;
12.
13.     public InputField r_username;
14.     public InputField r_pass;
15.
16.     private string username_r;
17.     private string pass_r;
18.
19.     public Text error;
20.
21.     private string g_string;
22.     private Color green = Color.blue;
23.     private Color red = new Color(64, 0, 0, 1);
24.
25.     void Start(){
26.         Button btn = submitBtn.GetComponent<Button>
        ();
    
```



```

27.         btn.onClick.AddListener(Submit);
28.     }
29.
30.     public void Submit()
31.     {
32.         error.text = "";
33.         username_r = r_username.text;
34.         pass_r = r_pass.text;
35.
36.         //error.text = username_r + pass_r;
37.         StartCoroutine(log_in(username_r, pass_r));
38.
39.         if (error.color == green)
40.         {
41.             PlayerPrefs.SetString("username", username_r);
42.             //Debug.Log(PlayerPrefs.GetString("username").ToString());
43.             Application.LoadLevel("pagePilihNomor");
44.         }
45.         else return;
46.     }
47.     IEnumerator log_in(string el, string el1)
48.     {
49.         UnityWebRequest link = UnityWebRequest.Get(
50.             "http://somethingnotright.dx.am/login2.php?r_username=" + el + "&r_pass=" + el1);
51.         yield return link.Send();
52.         if (link.isError)
53.         {
54.             Debug.Log(link.error);
55.         }
56.         else
57.         {
58.             g_string = link.downloadHandler.text;
59.             show_error(g_string.ToString());
60.         }
61.     }
62.     private void show_error(string el)
63.     {

```

```

64.         string[] arr = null;
65.         arr = el.Split(new string[] { " " }, StringSplitOptions.RemoveEmptyEntries);
66.         if (arr[2] == "Berhasil")
67.         {
68.             error.color = green;
69.             error.text = el;
70.         }
71.         else
72.         {
73.             error.color = red;
74.             error.text = el;
75.         }
76.     }
77. }

```

### Kode Sumber 4.3 Fungsi Login Pengguna

```

1. <?php
2.
3. header("Access-Control-Allow-Credentials: true");
4. header('Access-Control-Allow-Origin: *');
5. header('Access-Control-Allow-Methods: POST, GET, OPTIONS');
6. header('Access-Control-Allow-Headers: Accept, X-Access-Token, X-Application-Name, X-Request-Sent-Time');
7.
8. error_reporting(E_ALL ^ E_DEPRECATED);
9.
10. $Server = "fdb17.awardspace.net";
11. $User = "2326813_admin";
12. $Pass = "zzz1312000";
13. $Db = "2326813_admin";
14. $Connection = mysql_connect($Server, $User, $Pass, $Db);
15. if(mysql_error()){
16.     die("Connection Failed!". mysql_error());
17. }

```

```

18. mysql_select_db("2326813_admin", $Connection) or die
    e("Could not load to database" . mysql_error());
19.
20. $username = $_GET["r_username"];
21. $pass = md5(md5($_GET["r_pass"]));
22.
23. $check = mysql_query("SELECT * FROM peserta WHERE `
    username` = '". $username. "'");
24. $numrows = mysql_num_rows($check);
25.
26. if((empty($username))||(empty($pass))){
27.     echo "Data yang Anda masukkan kurang";
28. }
29.
30. else if($numrows == 0)
31. {
32.     die("Username yang Anda masukkan salah");
33. }
34.
35. else
36. {
37.     while($row = mysql_fetch_assoc($check))
38.     {
39.         if($pass == $row['password'])
40.         {
41.             die("Login Anda Berhasil");
42.         }
43.         else
44.         {
45.             die("Password yang anda masukkan salah"
46. );
47.         }
48.     }
49. }
50.
51. }
52.
53. ?>

```

#### Kode Sumber 4.4 Kontrol Login Pengguna

### 4.2.3 Implementasi Antarmuka Halaman Beranda

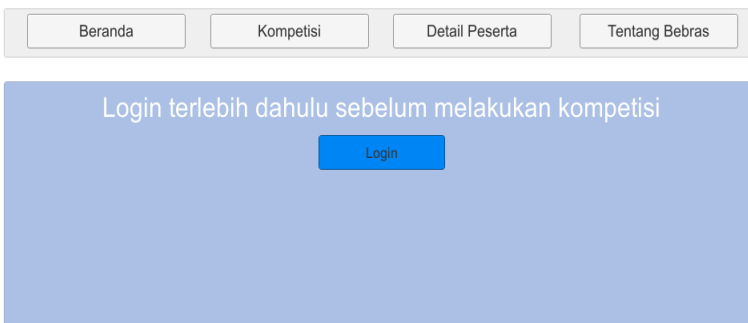
Antarmuka halaman Beranda pada Gambar 4.3 merupakan halaman yang akan muncul pertama kali saat membuka *web*.



**Gambar 4.3 Implementasi Halaman Antarmuka Beranda**

### 4.2.4 Implementasi Antarmuka Halaman Kompetisi

Antarmuka halaman Kompetisi pada Gambar 4.4 merupakan halaman untuk masuk kompetisi melalui *login* terlebih dahulu.



**Gambar 4.4 Implementasi Halaman Antarmuka Kompetisi**

### 4.2.5 Implementasi Antarmuka Halaman Detail Peserta

Antarmuka halaman Detail Peserta pada Gambar 4.5 merupakan halaman untuk melihat data peserta. Terdapat kode sumber fungsi detail peserta yang dapat dilihat pada Kode Sumber 4.5 dan kode sumber kontrol detail peserta pada Kode Sumber 4.6.



**Gambar 4.5 Implementasi Halaman Antarmuka Detail Peserta**

```

1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using UnityEngine.UI;
5.
6. public class DetailPeserta : MonoBehaviour {
7.
8.     public Text data;
9.     public Text ttl;
10.    public Text provin;
11.
12.    IEnumerator Start(){

```

```

13.         WWW itemsdata = new WWW("http://somethingno
    tright.dx.am/detailpeserta.php");
14.         WWW total = new WWW("http://somethingnotrig
    ht.dx.am/totalpeserta.php");
15.         WWW prov = new WWW("http://somethingnotrigh
    t.dx.am/detailprovinsi.php");
16.         yield return itemsdata;
17.         yield return total;
18.         yield return prov;
19.         string itemsdatastring = itemsdata.text;
20.         string itemstotal = total.text;
21.         string itemsprov = prov.text;
22.         print (itemsdatastring);
23.         print (itemstotal);
24.         print (itemsprov);
25.         data.text = itemsdatastring;
26.         ttl.text = itemstotal;
27.         provin.text = itemsprov;
28.     }
29. }

```

#### Kode Sumber 4.5 Fungsi Detail Peserta

```

1. <?php
2.
3. header("Access-Control-Allow-Credentials: true");
4. header('Access-Control-Allow-Origin: *');
5. header('Access-Control-Allow-
    Methods: POST, GET, OPTIONS');
6. header('Access-Control-Allow-Headers: Accept, X-
    Access-Token, X-Application-Name, X-Request-Sent-
    Time');
7.
8. error_reporting(E_ALL ^ E_DEPRECATED);
9.
10. $Server = "fdb17.awardspace.net";
11. $User = "2326813_admin";
12. $Pass = "zzz1312000";
13. $Db = "2326813_admin";

```

```

14. $Connection = mysql_connect($Server, $User, $Pass,
    $Db);
15.
16. if(mysql_error()){
17.     die("Connection Failed!=" . mysql_error());
18. }
19. mysql_select_db("2326813_admin", $Connection) or di
    e("Could not load to database" . mysql_error());
20.
21. $check = "SELECT username, name, provinsi FROM pese
    rta";
22. $result=mysql_query($check);
23. $numrows = mysql_num_rows($result);
24.
25. //while($row=mysql_fetch_array($result))
26. if($numrows > 0 ){
27.     while($row=mysql_fetch_array($result))
28.     {
29.         echo "\nUsername : ".$row['username']." ";
30.
31.         echo "(".$row['name'].") ";
32.         echo "- ".$row['provinsi']."\n";
33.     }
34. }
35. ?>

```

#### Kode Sumber 4.6 Kontrol Detail Peserta

```

1. <?php
2.
3. header("Access-Control-Allow-Credentials: true");
4.
5. header('Access-Control-Allow-Origin: *');
6.
7. header('Access-Control-Allow-
    Methods: POST, GET, OPTIONS');
8.
9. header('Access-Control-Allow-Headers: Accept, X-
    Access-Token, X-Application-Name, X-Request-Sent-
    Time');

```

```

10.
11. error_reporting(E_ALL ^ E_DEPRECATED);
12.
13. $Server = "fdb17.awardspace.net";
14.
15. $User = "2326813_admin";
16.
17. $Pass = "zzz1312000";
18.
19. $Db = "2326813_admin";
20.
21. $Connection = mysql_connect($Server, $User, $Pass,
    $Db);
22.
23. if(mysql_error()){
24.     die("Connection Failed!". mysql_error());
25. }
26.
27. mysql_select_db("2326813_admin", $Connection) or di
    e("Could not load to database" . mysql_error());
28.
29. $check = "SELECT COUNT(username) AS jumlah FROM pes
    erta";
30.
31. $result=mysql_query($check);
32. $numrows = mysql_num_rows($result);
33.
34. if($numrows > 0 ){
35.     while($row=mysql_fetch_array($result))
36.     {
37.         echo "Jumlah Peserta : ".$row['jumlah']."";
38.     }
39. }
40. ?>

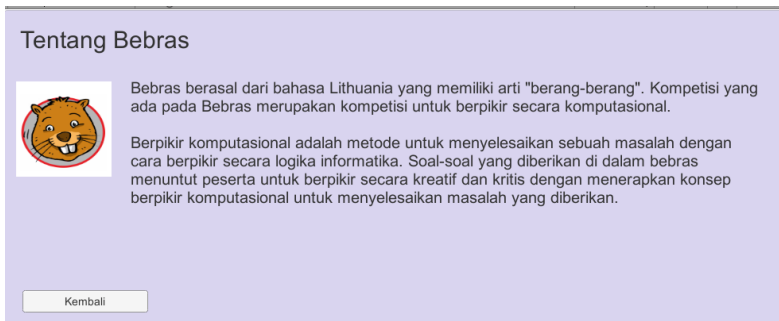
```

#### Kode Sumber 4.7 Kontrol Jumlah Peserta

### 4.2.6 Implementasi Halaman Antarmuka Tentang Bebras

Antarmuka halaman Tentang Bebras pada Gambar 4.6 merupakan halaman untuk melihat informasi tentang Bebras.





**Gambar 4.6 Implementasi Halaman Antarmuka Tentang Bebras**

#### **4.2.7 Implementasi Antarmuka Halaman Penjelasan Soal**

Antarmuka halaman Penjelasan Soal pada Gambar 4.7 merupakan halaman untuk melihat penjelasan soal.



**Gambar 4.7 Implementasi Halaman Antarmuka Penjelasan Soal**

#### **4.2.8 Implementasi Antarmuka Halaman Selesai**

Antarmuka halaman Selesai pada Gambar 4.8 merupakan halaman yang ditampilkan setelah selesai mengerjakan soal, berisi nilai yang didapatkan oleh peserta. Terdapat kode sumber fungsi

selesai yang dapat dilihat pada Kode Sumber 4.8 dan kode sumber kontrol selesai pada Kode Sumber 4.9.



**Gambar 4.8 Implementasi Antarmuka Halaman Selesai**

```

1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using UnityEngine.UI;
5. using System;
6. using UnityEngine.Networking;
7.
8. public class Selesai : MonoBehaviour {
9.
10.     public Text nilai;
11.     public Button submitBtn;
12.     string user;
13.     // Use this for initialization
14.     void Start()
15.     {
16.         Debug.Log(PlayerPrefs.GetString("username")
17.             .ToString());
18.         int score0 = PlayerPrefs.GetInt("scoreno0")
19.         ;
20.         int score1 = PlayerPrefs.GetInt("scoreno1")
21.         ;

```

```

19.         int score2 = PlayerPrefs.GetInt("scoreno2")
20.     ;
21.         int score3 = PlayerPrefs.GetInt("scoreno3")
22.     ;
23.         user = PlayerPrefs.GetString("username").ToString();
24.         int total = score0 + score1 + score2 + score3;
25.         print(total);
26.         PlayerPrefs.SetInt("TampilNilai", total);
27.         StartCoroutine(Postscore(user, total));
28.         Debug.Log(PlayerPrefs.GetInt("TampilNilai")
29.     );
30.         string Nilai = PlayerPrefs.GetInt("TampilNilai").ToString();
31.         nilai.text = Nilai;
32.         Button btn = submitBtn.GetComponent<Button>
33.     ();
34.         btn.onClick.AddListener(Submit);
35.     }
36.     public void Submit(){
37.         PlayerPrefs.DeleteAll();
38.         Application.LoadLevel ("pageHome");
39.     }
40.     IEnumerator Postscore(string user, int total)
41.     {
42.         UnityWebRequest link = UnityWebRequest.Get(
43.     "http://somethingnotright.dx.am/selesai.php?r_username=" + user + "&r_nilai=" + total);
44.         yield return link.Send();
45.         if (link.isError)
46.         {
47.             Debug.Log(link.error);
48.         }
49.     }
50.     // Update is called once per frame
51.     void Update () {
52.     }
53. }

```

### Kode Sumber 4.8 Fungsi Selesai

```

1. <?php
2. header("Access-Control-Allow-Credentials: true");
3. header('Access-Control-Allow-Origin: *');
4. header('Access-Control-Allow-
  Methods: POST, GET, OPTIONS');
5. header('Access-Control-Allow-Headers: Accept, X-
  Access-Token, X-Application-Name, X-Request-Sent-
  Time');
6.
7. error_reporting(E_ALL ^ E_DEPRECATED);
8. $Server = "fdb17.awardspace.net";
9. $User = "2326813_admin";
10. $Pass = "zzz1312000";
11. $Db = "2326813_admin";
12. $Connection = mysql_connect($Server, $User, $Pass,
  $Db);
13.
14. if(mysql_error()){
15.     die("Connection Failed!=". mysql_error());
16. }
17. mysql_select_db("2326813_admin", $Connection) or di
  e("Could not load to database" . mysql_error());
18.
19. $username = $_GET["r_username"];
20. $nilai = $_GET["r_nilai"];
21.
22. $check = mysql_query("SELECT * FROM peserta WHERE `
  username` = '". $username.'");
23. $numrows = mysql_num_rows($check);
24.
25. if($numrows == 0)
26. {
27.     die("Error");
28. }
29. else
30. {
31.     while($row = mysql_fetch_assoc($check))
32.     {
33.         if($username == $row['username'])
34.         {

```

```

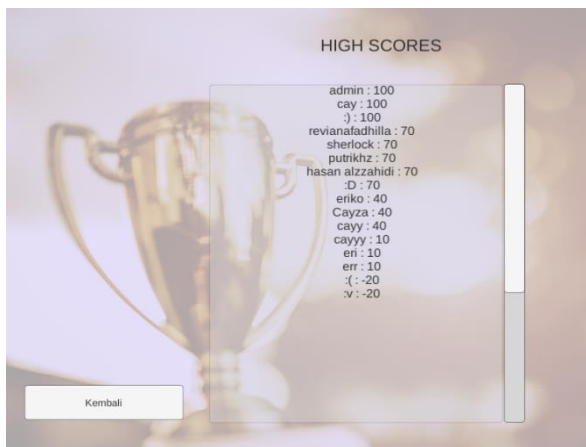
35.         $sql = mysql_query("UPDATE `peserta` SE
T `nilai` = ".$nilai." WHERE `username` = ".$use
rname."");
36.     }
37.     else
38.     {
39.         die("error");
40.     }
41. }
42. }
43. ?>

```

#### Kode Sumber 4.9 Kontrol Fungsi Selesai

### 4.2.9 Implementasi Antarmuka Halaman Peringkat

Antarmuka halaman Peringkat pada Gambar 4.9 merupakan halaman yang berisi data semua nilai peserta. Terdapat kode sumber fungsi peringkat yang dapat dilihat pada Kode Sumber 4.10 dan kode sumber kontrol peringkat pada Kode Sumber 4.11.



**Gambar 4.9 Implementasi Antarmuka Halaman Peringkat**

```

1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using UnityEngine.UI;
5.
6. public class Ranking : MonoBehaviour {
7.     public Text data;
8.     IEnumerator Start(){
9.         WWW itemsdata = new WWW("http://somethingno
tright.dx.am/ranking.php");
10.        yield return itemsdata;
11.        string itemsdatastring = itemsdata.text;
12.        print (itemsdatastring);
13.        data.text = itemsdatastring;
14.    }
15. }

```

#### Kode Sumber 4.10 Fungsi Peringkat

```

1. <?php
2.
3. header("Access-Control-Allow-Credentials: true");
4. header('Access-Control-Allow-Origin: *');
5. header('Access-Control-Allow-
Methods: POST, GET, OPTIONS');
6. header('Access-Control-Allow-Headers: Accept, X-
Access-Token, X-Application-Name, X-Request-Sent-
Time');
7.
8. error_reporting(E_ALL ^ E_DEPRECATED);
9.
10. $Server = "fdb17.awardspace.net";
11. $User = "2326813_admin";
12. $Pass = "zzz1312000";
13. $Db = "2326813_admin";
14. $Connection = mysql_connect($Server, $User, $Pass,
$Db);
15.
16. if(mysql_error()){
17.     die("Connection Failed!=". mysql_error());

```

```

18. }
19. mysql_select_db("2326813_admin", $Connection) or die("Could not load to database" . mysql_error());
20.
21. $check = "SELECT username, nilai FROM peserta ORDER BY nilai DESC";
22. $result=mysql_query($check);
23. $numrows = mysql_num_rows($result);
24.
25. if($numrows > 0 ){
26.     while($row=mysql_fetch_array($result))
27.     {
28.         echo " " . $row['username']. " : " . $row['nilai']. "\n" ;
29.     }
30. }
31. ?>

```

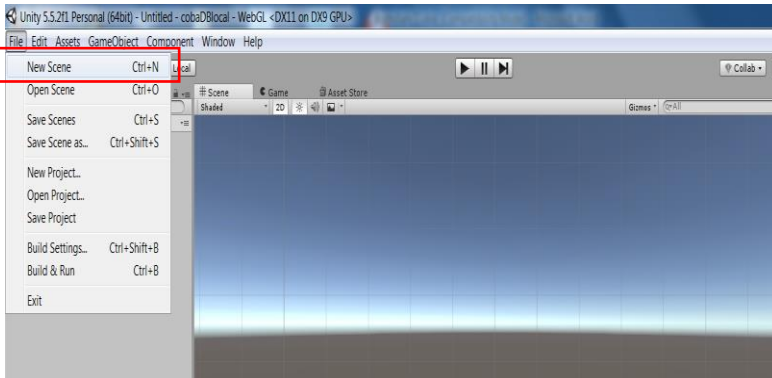
**Kode Sumber 4.11 Kontrol Fungsi Peringkat**

### 4.3 Implementasi Pembuatan Aplikasi

Pada tahap implementasi pembuatan aplikasi ini akan membahas tentang implementasi pembuatan *scene*, implementasi import *assets*, implementasi *load* objek ke *scene*, serta implementasi pembuatan *script*.

#### 4.3.1 Implementasi Pembuatan *Scene*

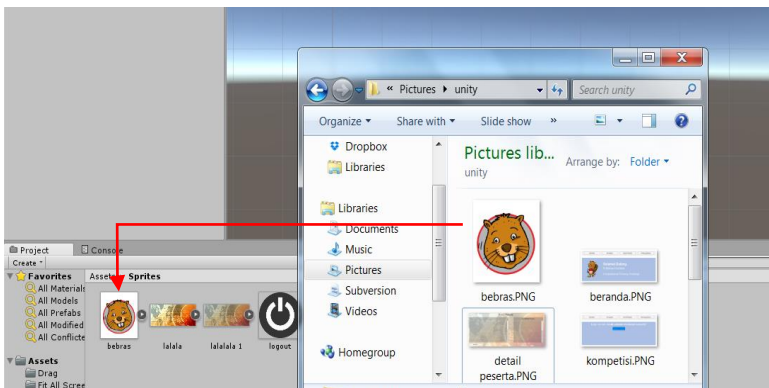
Untuk menambahkan *scene*, hal yang perlu dilakukan adalah menuju menu “File” lalu pilih “New Scene”. Setelah *scene* baru terbuat, kita dapat menambah objek ke dalam *scene* dan mengatur sesuai rancangan yang dibuat seperti yang dapat dilihat pada Gambar 4.10



**Gambar 4.10 Pembuatan Scene**

### 4.3.2 Implementasi *Import Assets*

Implementasi *import assets* dapat dilakukan dengan cara *drag and drop* dari *file explore* ke “Project” di dalam folder “Assets” pada unity. Lebih jelasnya dapat dilihat pada Gambar 4.11. Kita juga dapat mengatur pengelompokan *asset* dengan membuat *folder* supaya lebih rapi dan memudahkan dalam pencarian *asset*.

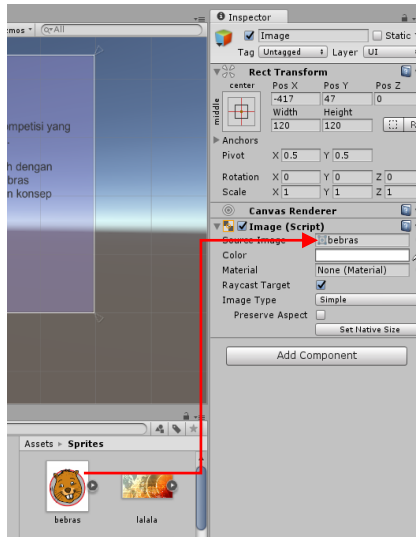


**Gambar 4.11 Import Asset ke Proyek Unity**



### 4.3.3 Implementasi *Load* Objek ke *Scene*

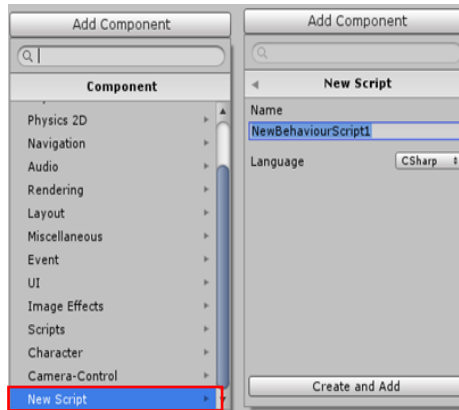
Untuk memasukan objek-objek ke dalam *scene* dapat dimasukkan dengan cara *drag and drop* berkas dari tab “Project” ke dalam *scene*. Lebih jelasnya dapat dilihat pada Gambar 4.12.



Gambar 4.12 Load Objek ke *Scene*

### 4.3.4 Implementasi Pembuatan *Script*

Untuk membuat *script* yang mempengaruhi objek adalah dengan memilih terlebih dahulu objek yang akan diberi *script*. Pada tab “Inspector” tambahkan komponen *script*. Jika tidak ada *script* yang sesuai, kita akan membuat *file script* baru dan mengubah isi *file* sesuai yang kita inginkan seperti pada Gambar 4.13.



**Gambar 4.13 Pembuatan Script Baru**

## 4.3.5 Implementasi Pembuatan Soal

### 4.3.5.1 Implementasi Pembuatan Soal Satu

Soal satu mengimplementasikan fungsi *shortest path* yaitu mencari jarak terdekat. Terdapat banyak jalan berupa tombol yang memiliki nilai benar dan salah. Jika peserta menekan jalan yang memiliki nilai benar semua maka jawaban peserta tersebut adalah benar. Inisiasi variabel jalan dapat dilihat pada Kode sumber 4.12. Disetiap jalan memiliki fungsi masing-masing. Fungsi untuk jalan yang benar dapat dilihat pada Kode Sumber 4.13, Kode Sumber 4.14, Kode Sumber 4.15, Kode Sumber 4.16, Kode Sumber 4.17, Kode Sumber 4.18, Kode Sumber 4.19, Kode Sumber 4.28, Kode Sumber 4.29, Kode Sumber 4.33, dan Kode Sumber 4.34. Fungsi untuk jalan yang salah dapat dilihat pada Kode Sumber 4.20, Kode Sumber 4.21 Kode Sumber 4.22, Kode Sumber 4.23, Kode Sumber 4.24, Kode Sumber 4.25, Kode Sumber 4.26, Kode Sumber 4.27, Kode Sumber 4.30, Kode Sumber 4.31, Kode Sumber 4.32, Kode Sumber 4.33, Kode Sumber 4.34, Kode Sumber 4.35, Kode Sumber 4.36, Kode Sumber 4.37, Kode Sumber 4.38, dan Kode Sumber 4.39.

```

1.  public Button mybutton, mybutton2, mybutton3, mybutton4, mybutton5, mybutton6, mybutton7, mybutton8, mybutton9, mybutton10,
2.      mybutton11, mybutton12, mybutton13, mybutton14, mybutton15, mybutton16, mybutton17, mybutton18, mybutton19, mybutton20,
3.      mybutton21, mybutton22, mybutton23, mybutton24, mybutton25, mybutton26, mybutton27;
4.      public Sprite jalan;
5.      public Sprite pencet;
6.      private int counter1, counter2, counter3, counter4, counter5, counter6, counter7, counter8, counter9, counter10,
7.          counter11, counter12, counter13, counter14, counter15, counter16, counter17, counter18, counter19, counter20,
8.          counter21, counter22, counter23, counter24, counter25, counter26, counter27 = 0;

```

#### Kode Sumber 4.12 Inisiasi Variabel Jalan

```

1.
2.  public void jalan1()
3.  {
4.      counter1++;
5.      if (counter1 % 2 == 0)
6.      {
7.          mybutton.image.overrideSprite = jalan;
8.          PlayerPrefs.SetInt("jalan1", 0);
9.          Debug.Log(PlayerPrefs.GetInt("jalan1"))
10.         ;
11.     } else
12.     {
13.         mybutton.image.overrideSprite = pencet;
14.         PlayerPrefs.SetInt("jalan1", 1);
15.         Debug.Log(PlayerPrefs.GetInt("jalan1"))
16.         ;
17.     }
18. }

```

#### Kode Sumber 4.13 Fungsi Jalan 1

```

1.  public void jalan2()
2.  {
3.      counter2++;
4.      if (counter2 % 2 == 0)
5.      {
6.          mybutton2.image.overrideSprite = jalan;
7.
8.          PlayerPrefs.SetInt("jalan2", 0);
9.          Debug.Log(PlayerPrefs.GetInt("jalan2"))
10.         ;
11.     }
12.     else
13.     {
14.         mybutton2.image.overrideSprite = pencet
15.
16.         PlayerPrefs.SetInt("jalan2", 1);
17.         Debug.Log(PlayerPrefs.GetInt("jalan2"))
18.
19.     ;
20.     }
21. }

```

#### Kode Sumber 4.14 Fungsi Jalan 2

```

1.  public void jalan3()
2.  {
3.      counter3++;
4.      if (counter3 % 2 == 0)
5.      {
6.          mybutton3.image.overrideSprite = jalan;
7.
8.          PlayerPrefs.SetInt("jalan3", 0);
9.          Debug.Log(PlayerPrefs.GetInt("jalan3"))
10.         ;
11.     }
12.     else
13.     {
14.         mybutton3.image.overrideSprite = pencet
15.
16.         PlayerPrefs.SetInt("jalan3", 1);

```

```

15.         Debug.Log(PlayerPrefs.GetInt("jalan3"))
16.     ;
17.     }
18. }

```

#### Kode Sumber 4.15 Fungsi Jalan 3

```

1.     public void jalan4()
2.     {
3.         counter4++;
4.         if (counter4 % 2 == 0)
5.         {
6.             mybutton4.image.overrideSprite = jalan;
7.
8.             PlayerPrefs.SetInt("jalan4", 0);
9.             Debug.Log(PlayerPrefs.GetInt("jalan4"))
10.        ;
11.        }
12.        else
13.        {
14.            mybutton4.image.overrideSprite = pencet
15.        ;
16.            PlayerPrefs.SetInt("jalan4", 1);
17.            Debug.Log(PlayerPrefs.GetInt("jalan4"))
18.        ;
19.        }
20.    }

```

#### Kode Sumber 4.16 Fungsi Jalan 4

```

1.     public void jalan5()
2.     {
3.         counter5++;
4.         if (counter5 % 2 == 0)
5.         {
6.             mybutton5.image.overrideSprite = jalan;
7.
8.             PlayerPrefs.SetInt("jalan5", 0);

```

```

8.         Debug.Log(PlayerPrefs.GetInt("jalan5"))
9.     ;
10.    }
11.    else
12.    {
13.        mybutton5.image.overrideSprite = pencet
14.    ;
15.        PlayerPrefs.SetInt("jalan5", 1);
16.        Debug.Log(PlayerPrefs.GetInt("jalan5"))
17.    ;
18.    }

```

**Kode Sumber 4.17 Fungsi Jalan 5**

```

1.    public void jalan6()
2.    {
3.        counter6++;
4.        if (counter6 % 2 == 0)
5.        {
6.            mybutton6.image.overrideSprite = jalan;
7.
8.            PlayerPrefs.SetInt("jalan6", 0);
9.            Debug.Log(PlayerPrefs.GetInt("jalan6"))
10.        ;
11.        }
12.    else
13.    {
14.        mybutton6.image.overrideSprite = pencet
15.    ;
16.        PlayerPrefs.SetInt("jalan6", 1);
17.        Debug.Log(PlayerPrefs.GetInt("jalan6"))
18.    ;
19.    }
20. }

```

**Kode Sumber 4.18 Fungsi Jalan 6**

```

1.  public void jalan7()
2.  {
3.      counter7++;
4.      if (counter7 % 2 == 0)
5.      {
6.          mybutton7.image.overrideSprite = jalan;
7.
8.          PlayerPrefs.SetInt("jalan7", 0);
9.          Debug.Log(PlayerPrefs.GetInt("jalan7"))
10.         ;
11.     }
12.     else
13.     {
14.         mybutton7.image.overrideSprite = pencet
15.         ;
16.         PlayerPrefs.SetInt("jalan7", 1);
17.         Debug.Log(PlayerPrefs.GetInt("jalan7"))
18.         ;
19.     }
20. }

```

**Kode Sumber 4.19 Fungsi Jalan 7**

```

1.
2.  public void jalan8()
3.  {
4.      counter8++;
5.      if (counter8 % 2 == 0)
6.      {
7.          mybutton8.image.overrideSprite = jalan;
8.
9.          PlayerPrefs.SetInt("jalan8", 0);
10.         Debug.Log(PlayerPrefs.GetInt("jalan8"))
11.         ;
12.     }
13.     else
14.     {
15.         mybutton8.image.overrideSprite = pencet
16.         ;
17.         PlayerPrefs.SetInt("jalan8", 20);

```

```

15.             Debug.Log(PlayerPrefs.GetInt("jalan8"))
16.         ;
17.     }
18.

```

#### Kode Sumber 4.20 Fungsi Jalan 8

```

1.  public void jalan9()
2.  {
3.      counter9++;
4.      if (counter9 % 2 == 0)
5.      {
6.          mybutton9.image.overrideSprite = jalan;
7.
8.          PlayerPrefs.SetInt("jalan9", 0);
9.          Debug.Log(PlayerPrefs.GetInt("jalan9"))
10.         ;
11.     }
12.     else
13.     {
14.         mybutton9.image.overrideSprite = pencet
15.         ;
16.         PlayerPrefs.SetInt("jalan9", 20);
17.         Debug.Log(PlayerPrefs.GetInt("jalan9"))
18.         ;
19.     }
20. }

```

#### Kode Sumber 4.21 Fungsi Jalan 9

```

1.  public void jalan10()
2.  {
3.      counter10++;
4.      if (counter10 % 2 == 0)
5.      {
6.          mybutton10.image.overrideSprite = jalan
7.          ;
8.          PlayerPrefs.SetInt("jalan10", 0);
9.          Debug.Log(PlayerPrefs.GetInt("jalan10"))
10.         );
11.     }
12. }

```



```

10.         }
11.         else
12.         {
13.             mybutton10.image.overrideSprite = pence
14.             t;
15.             PlayerPrefs.SetInt("jalan10", 20);
16.             Debug.Log(PlayerPrefs.GetInt("jalan10"))
17.         );
18.     }

```

**Kode Sumber 4.22 Fungsi Jalan 10**

```

1.  public void jalan11()
2.  {
3.      {
4.          counter11++;
5.          if (counter11 % 2 == 0)
6.          {
7.              mybutton11.image.overrideSprite = jalan
8.              ;
9.              PlayerPrefs.SetInt("jalan11", 0);
10.             Debug.Log(PlayerPrefs.GetInt("jalan11"))
11.         );
12.     }
13.     }
14.     else
15.     {
16.         mybutton11.image.overrideSprite = pence
17.         t;
18.         PlayerPrefs.SetInt("jalan11", 20);
19.         Debug.Log(PlayerPrefs.GetInt("jalan11"))
20.     );
21.     }
22. }

```

**Kode Sumber 4.23 Fungsi Jalan 11**

```

1.  public void jalan12()
2.      {
3.          counter12++;
4.          if (counter12 % 2 == 0)
5.          {
6.              mybutton12.image.overrideSprite = jalan
7.          ;
8.              PlayerPrefs.SetInt("jalan12", 0);
9.              Debug.Log(PlayerPrefs.GetInt("jalan12"))
10.         );
11.         } else
12.         {
13.             mybutton12.image.overrideSprite = pence
14.         t;
15.             PlayerPrefs.SetInt("jalan12", 20);
16.             Debug.Log(PlayerPrefs.GetInt("jalan12"))
17.         );
18.         }
19.     }
20. }

```

**Kode Sumber 4.24 Fungsi Jalan 12**

```

1.  public void jalan13()
2.      {
3.          counter13++;
4.          if (counter13 % 2 == 0)
5.          {
6.              mybutton13.image.overrideSprite = jalan
7.          ;
8.              PlayerPrefs.SetInt("jalan13", 0);
9.              Debug.Log(PlayerPrefs.GetInt("jalan13"))
10.         );
11.         } else {
12.             mybutton13.image.overrideSprite = pence
13.         t;
14.             PlayerPrefs.SetInt("jalan13", 20);
15.             Debug.Log(PlayerPrefs.GetInt("jalan13"))
16.         ); }
17.     }

```

**Kode Sumber 4.25 Fungsi Jalan 13**

```

1.  public void jalan14()
2.  {
3.      counter14++;
4.      if (counter14 % 2 == 0)
5.      {
6.          mybutton14.image.overrideSprite = jalan
7.      ;
8.          PlayerPrefs.SetInt("jalan14", 0);
9.          Debug.Log(PlayerPrefs.GetInt("jalan14"))
10.     );
11.     } else
12.     {
13.         mybutton14.image.overrideSprite = pence
14.     t;
15.         PlayerPrefs.SetInt("jalan14", 20);
16.         Debug.Log(PlayerPrefs.GetInt("jalan14"))
17.     );
18.     }
19. }

```

#### Kode Sumber 4.26 Fungsi Jalan 14

```

1.  public void jalan15()
2.  {
3.      counter15++;
4.      if (counter15 % 2 == 0)
5.      {
6.          mybutton15.image.overrideSprite = jalan
7.      ;
8.          PlayerPrefs.SetInt("jalan15", 0);
9.          Debug.Log(PlayerPrefs.GetInt("jalan15"))
10.     );
11.     } else
12.     {
13.         mybutton15.image.overrideSprite = pence
14.     t;
15.         PlayerPrefs.SetInt("jalan15", 20);

```

```

14.         Debug.Log(PlayerPrefs.GetInt("jalan15"))
15.     );
16.     }
17. }

```

#### Kode Sumber 4.27 Fungsi Jalan 15

```

1.  public void jalan16()
2.  {
3.      counter16++;
4.      if (counter16 % 2 == 0)
5.      {
6.          mybutton16.image.overrideSprite = jalan
7.          ;
8.          PlayerPrefs.SetInt("jalan16", 0);
9.          Debug.Log(PlayerPrefs.GetInt("jalan16"))
10.     );
11.     } else
12.     {
13.         mybutton16.image.overrideSprite = pence
14.         t;
15.         PlayerPrefs.SetInt("jalan16", 1);
16.         Debug.Log(PlayerPrefs.GetInt("jalan16"))
17.     );
18.     }
19. }

```

#### Kode Sumber 4.28 Fungsi Jalan 16

```

1.  public void jalan17()
2.  {
3.      counter17++;
4.      if (counter17 % 2 == 0)
5.      {
6.          mybutton17.image.overrideSprite = jalan
7.          ;
8.          PlayerPrefs.SetInt("jalan17", 0);
9.          Debug.Log(PlayerPrefs.GetInt("jalan17"))
10.     );
11.     }
12. }

```

```

9.
10.         } else
11.         {
12.             mybutton17.image.overrideSprite = pence
13.             t;
14.             PlayerPrefs.SetInt("jalan17", 1);
15.             Debug.Log(PlayerPrefs.GetInt("jalan17"))
16.         );
17.     }

```

**Kode Sumber 4.29 Fungsi Jalan 17**

```

1.     public void jalan18()
2.     {
3.         counter18++;
4.         if (counter18 % 2 == 0)
5.         {
6.             mybutton18.image.overrideSprite = jalan
7.             ;
8.             PlayerPrefs.SetInt("jalan18", 0);
9.             Debug.Log(PlayerPrefs.GetInt("jalan18"))
10.        );
11.        } else
12.        {
13.            mybutton18.image.overrideSprite = pence
14.            t;
15.            PlayerPrefs.SetInt("jalan18", 20);
16.            Debug.Log(PlayerPrefs.GetInt("jalan18"))
17.        );
18.    }

```

**Kode Sumber 4.30 Fungsi Jalan 18**

```

1.     public void jalan19()
2.     {
3.         counter19++;
4.         if (counter19 % 2 == 0)

```

```

5.         {
6.             mybutton19.image.overrideSprite = jalan
7.         ;
8.             PlayerPrefs.SetInt("jalan19", 0);
9.             Debug.Log(PlayerPrefs.GetInt("jalan19")
10.        );
11.        } else
12.        {
13.            mybutton19.image.overrideSprite = pence
14.        t;
15.            PlayerPrefs.SetInt("jalan19", 20);
16.            Debug.Log(PlayerPrefs.GetInt("jalan19")
17.        );
18.        }
19.    }
20. }

```

**Kode Sumber 4.31 Fungsi Jalan 19**

```

1. public void jalan20()
2. {
3.     counter20++;
4.     if (counter20 % 2 == 0)
5.     {
6.         mybutton20.image.overrideSprite = jalan
7.     ;
8.         PlayerPrefs.SetInt("jalan20", 0);
9.         Debug.Log(PlayerPrefs.GetInt("jalan20")
10.    );
11.    } else
12.    {
13.        mybutton20.image.overrideSprite = pence
14.    t;
15.        PlayerPrefs.SetInt("jalan20", 20);
16.        Debug.Log(PlayerPrefs.GetInt("jalan20")
17.    );
18.    }
19. }

```

**Kode Sumber 4.32 Fungsi Jalan 20**

```

1.     public void jalan21()
2.     {
3.         counter21++;
4.         if (counter21 % 2 == 0)
5.         {
6.             mybutton21.image.overrideSprite = jalan
7.         ;
8.             PlayerPrefs.SetInt("jalan21", 0);
9.             Debug.Log(PlayerPrefs.GetInt("jalan21"))
10.        );
11.        } else
12.        {
13.            mybutton21.image.overrideSprite = pence
14.        t;
15.            PlayerPrefs.SetInt("jalan21", 1);
16.            Debug.Log(PlayerPrefs.GetInt("jalan21"))
17.        );
18.        }
19.    }

```

#### Kode Sumber 4.33 Fungsi Jalan 21

```

1.     public void jalan22()
2.     {
3.         counter22++;
4.         if (counter22 % 2 == 0)
5.         {
6.             mybutton22.image.overrideSprite = jalan
7.         ;
8.             PlayerPrefs.SetInt("jalan22", 0);
9.             Debug.Log(PlayerPrefs.GetInt("jalan22"))
10.        );
11.        } else
12.        {
13.            mybutton22.image.overrideSprite = pence
14.        t;

```

```

14.         PlayerPrefs.SetInt("jalan22", 1);
15.         Debug.Log(PlayerPrefs.GetInt("jalan22")
16.     );
17.
18.     }
19. }

```

**Kode Sumber 4.34 Fungsi Jalan 22**

```

1.     public void jalan23()
2.     {
3.         counter23++;
4.         if (counter23 % 2 == 0)
5.         {
6.             mybutton23.image.overrideSprite = jalan
7.         ;
8.             PlayerPrefs.SetInt("jalan23", 0);
9.             Debug.Log(PlayerPrefs.GetInt("jalan23")
10.        );
11.        } else
12.        {
13.            mybutton23.image.overrideSprite = pence
14.        t;
15.            PlayerPrefs.SetInt("jalan23", 20);
16.            Debug.Log(PlayerPrefs.GetInt("jalan23")
17.        );
18.        }
19.    }

```

**Kode Sumber 4.35 Fungsi Jalan 23**

```

1.     public void jalan24()
2.     {
3.         counter24++;
4.         if (counter24 % 2 == 0)
5.         {
6.             mybutton24.image.overrideSprite = jalan
7.         ;
8.             PlayerPrefs.SetInt("jalan24", 0);

```



```

8.         Debug.Log(PlayerPrefs.GetInt("jalan24"))
9.     );
10.
11.     } else
12.     {
13.         mybutton24.image.overrideSprite = pence
14.         t;
15.         PlayerPrefs.SetInt("jalan24", 20);
16.         Debug.Log(PlayerPrefs.GetInt("jalan24"))
17.     );
18.     }

```

#### Kode Sumber 4.36 Fungsi Jalan 24

```

1.  public void jalan25()
2.  {
3.      counter25++;
4.      if (counter25 % 2 == 0)
5.      {
6.          mybutton25.image.overrideSprite = jalan
7.          ;
8.          PlayerPrefs.SetInt("jalan25", 0);
9.          Debug.Log(PlayerPrefs.GetInt("jalan25"))
10.     );
11.     } else
12.     {
13.         mybutton25.image.overrideSprite = pence
14.         t;
15.         PlayerPrefs.SetInt("jalan25", 20);
16.         Debug.Log(PlayerPrefs.GetInt("jalan25"))
17.     );
18.     }

```

#### Kode Sumber 4.37 Fungsi Jalan 25

```

1.  public void jalan26()
2.  {
3.      counter26++;
4.      if (counter26 % 2 == 0)
5.      {
6.          mybutton26.image.overrideSprite = jalan
7.      ;
8.          PlayerPrefs.SetInt("jalan26", 0);
9.          Debug.Log(PlayerPrefs.GetInt("jalan26"))
10.     );
11.     } else
12.     {
13.         mybutton26.image.overrideSprite = pence
14.     t;
15.         PlayerPrefs.SetInt("jalan26", 20);
16.         Debug.Log(PlayerPrefs.GetInt("jalan26"))
17.     );
18.     }
19. }

```

**Kode Sumber 4.38 Fungsi Jalan 26**

```

1.  public void jalan27()
2.  {
3.      counter27++;
4.      if (counter27 % 2 == 0) {
5.          mybutton27.image.overrideSprite = jalan
6.      ;
7.          PlayerPrefs.SetInt("jalan27", 0);
8.          Debug.Log(PlayerPrefs.GetInt("jalan27"))
9.      );
10.     } else
11.     {
12.         mybutton27.image.overrideSprite = pence
13.     t;
14.         PlayerPrefs.SetInt("jalan27", 20);
15.         Debug.Log(PlayerPrefs.GetInt("jalan27"))
16.     ); }
17. }

```

**Kode Sumber 4.39 Fungsi Jalan 27**

### 4.3.5.2 Implementasi Pembuatan Soal Dua

Soal dua mengimplementasikan fungsi dari *decision tree* dimana peserta harus menjawab berdasarkan aturan *decision tree*. Terdapat 3 pilihan jawaban bergambar rumah, jika peserta memilih rumah yang sesuai dengan pertanyaan maka jawaban peserta tersebut adalah benar. Inisiasi variabel rumah dapat dilihat pada Kode Sumber 4.40. Fungsi untuk pilihan rumah yang benar dapat dilihat pada Kode Sumber 4.43, fungsi untuk pilihan rumah yang salah dapat dilihat pada Kode Sumber 4.41 dan Kode Sumber 4.42.

```

1. public Button mybutton1, mybutton2, mybutton3;
2. public Sprite rumah1;
3. public Sprite rumah2;
4. public Sprite rumah3;
5. public Sprite rumah1dipilih;
6. public Sprite rumah2dipilih;
7. public Sprite rumah3dipilih;
8. private int counter1, counter2, counter3 = 0;

```

**Kode Sumber 4.40 Inisiasi Variabel Rumah**

```

1. public void rmh1()
2. {
3.     counter1++;
4.     if (counter1 % 2 == 0)
5.     {
6.         mybutton1.image.overrideSprite = rumah1
7.         ;
8.         PlayerPrefs.SetInt("rumah1", 0);
9.         Debug.Log(PlayerPrefs.GetInt("rumah2"))
10.        ; } else {
11.        mybutton1.image.overrideSprite = rumah1
12.        dipilih;
13.        PlayerPrefs.SetInt("rumah1", 0);
14.        Debug.Log (PlayerPrefs.GetInt ("rumah1"
15.        )); }
16.    }

```

**Kode Sumber 4.41 Fungsi Rumah 1**

```

1.  public void rmh2()
2.  {
3.      counter2++;
4.      if (counter2 % 2 == 0)
5.      {
6.          mybutton2.image.overrideSprite = rumah2
7.      ;
8.          PlayerPrefs.SetInt("rumah2", 0);
9.          Debug.Log(PlayerPrefs.GetInt("rumah2"))
10.     ;
11.     } else
12.     {
13.         mybutton2.image.overrideSprite = rumah2
14.     dipilih;
15.         PlayerPrefs.SetInt("rumah2", 0);
16.         Debug.Log (PlayerPrefs.GetInt ("rumah2"
17.     ));
18.     }
19. }

```

**Kode Sumber 4.42 Fungsi Rumah 2**

```

1.  public void rmh3()
2.  {
3.      counter3++;
4.      if (counter3 % 2 == 0)
5.      {
6.          mybutton3.image.overrideSprite = rumah3
7.      ;
8.          PlayerPrefs.SetInt("rumah3", 0);
9.          Debug.Log(PlayerPrefs.GetInt("rumah3"))
10.     ; } else
11.     {
12.         mybutton3.image.overrideSprite = rumah3
13.     dipilih;
14.         PlayerPrefs.SetInt("rumah3", 1);
15.         Debug.Log (PlayerPrefs.GetInt ("rumah3"
16.     )); }
17. }

```

**Kode Sumber 4.43 Fungsi Rumah 3**

### 4.3.5.3 Impelentasi Pembuatan Soal Tiga

Soal tiga mengimplementasikan fungsi *shortest path* yaitu mencari jarak terdekat. Terdapat banyak jalan berupa tombol yang memiliki nilai benar dan salah. Jika peserta menekan jalan yang memiliki nilai benar semua maka jawaban peserta tersebut adalah benar. Inisiasi variabel jalan dapat dilihat pada Kode Sumber 4.44. Disetiap jalan memiliki fungsi masing-masing. Fungsi untuk jalan yang benar dapat dilihat pada Kode Sumber 4.48, Kode Sumber 4.49, dan Kode Sumber 4.50, fungsi untuk jalan yang salah dapat dilihat pada Kode Sumber 4.45, Kode Sumber 4.46, Kode Sumber 4.47, Kode Sumber 4.51, Kode Sumber 4.52, dan Kode Sumber 4.53.

```

1.  public Button mybutton, mybutton2, mybutton3, myb
   utton4, mybutton5, mybutton6, mybutton7, mybutton8,
   mybutton9;
2.  public Sprite jalan;
3.  public Sprite pencet;
4.  private int counter1, counter2, counter3, count
   er4, counter5, counter6, counter7, counter8, counter
   9 = 0;

```

#### Kode Sumber 4.44 Inisiasi Variabel Jalan

```

1.  public void jalan1()
2.  {
3.      counter1++;
4.      if (counter1 % 2 == 0)
5.      {
6.          mybutton.image.overrideSprite = jalan;
7.
8.          PlayerPrefs.SetInt("2jalan1", 0);
9.          Debug.Log(PlayerPrefs.GetInt("2jalan1")
10. );
11.     }
12.     else
13.     {
14.         mybutton.image.overrideSprite = pencet;

```

```

13.             PlayerPrefs.SetInt("2jalan1", 20);
14.             Debug.Log(PlayerPrefs.GetInt("2jalan1")
15. );
16.     }

```

#### Kode Sumber 4.45 Fungsi Jalan 1

```

1.  public void jalan2()
2.  {
3.      counter2++;
4.      if (counter2 % 2 == 0)
5.      {
6.          mybutton2.image.overrideSprite = jalan;
7.
8.          PlayerPrefs.SetInt("2jalan2", 0);
9.          Debug.Log(PlayerPrefs.GetInt("2jalan2")
10. );
11.      }
12.      else
13.      {
14.          mybutton2.image.overrideSprite = pencet
15. ;
16.          PlayerPrefs.SetInt("2jalan2", 20);
17.          Debug.Log(PlayerPrefs.GetInt("2jalan2")
18. );
19.      }
20. }

```

#### Kode Sumber 4.46 Fungsi Jalan 2

```

1.  public void jalan3()
2.  {
3.      counter3++;
4.      if (counter3 % 2 == 0)
5.      {
6.          mybutton3.image.overrideSprite = jalan;
7.
8.          PlayerPrefs.SetInt("2jalan3", 0);
9.          Debug.Log(PlayerPrefs.GetInt("2jalan3")
10. );

```

```

9.         }
10.        else
11.        {
12.            mybutton3.image.overrideSprite = pencet
13.        ;
14.            PlayerPrefs.SetInt("2jalan3", 20);
15.            Debug.Log(PlayerPrefs.GetInt("2jalan3")
16.        );
17.        }

```

**Kode Sumber 4.47 Fungsi Jalan 3**

```

1.    public void jalan4()
2.    {
3.        counter4++;
4.        if (counter4 % 2 == 0)
5.        {
6.            mybutton4.image.overrideSprite = jalan;
7.
8.            PlayerPrefs.SetInt("2jalan4", 0);
9.            Debug.Log(PlayerPrefs.GetInt("2jalan4")
10.        );
11.        }
12.        else
13.        {
14.            mybutton4.image.overrideSprite = pencet
15.        ;
16.            PlayerPrefs.SetInt("2jalan4", 1);
17.            Debug.Log(PlayerPrefs.GetInt("2jalan4")
18.        );
19.        }
20.    }

```

**Kode Sumber 4.48 Fungsi Jalan 4**

```

1.    public void jalan5()
2.    {
3.        counter5++;
4.        if (counter5 % 2 == 0)
5.        {

```

```

6.          mybutton5.image.overrideSprite = jalan;
7.          PlayerPrefs.SetInt("2jalan5", 0);
8.          Debug.Log(PlayerPrefs.GetInt("2jalan5")
9.      );
10.         }
11.         else
12.         {
13.             mybutton5.image.overrideSprite = pencet
14.         ;
15.             PlayerPrefs.SetInt("2jalan5", 1);
16.             Debug.Log(PlayerPrefs.GetInt("2jalan5")
17.         );
18.         }
19.     }

```

**Kode Sumber 4.49 Fungsi Jalan 5**

```

1.     public void jalan6()
2.     {
3.         counter6++;
4.         if (counter6 % 2 == 0)
5.         {
6.             mybutton6.image.overrideSprite = jalan;
7.             PlayerPrefs.SetInt("2jalan6", 0);
8.             Debug.Log(PlayerPrefs.GetInt("2jalan6")
9.         );
10.        }
11.        else
12.        {
13.            mybutton6.image.overrideSprite = pencet
14.        ;
15.            PlayerPrefs.SetInt("2jalan6", 1);
16.            Debug.Log(PlayerPrefs.GetInt("2jalan6")
17.        );
18.        }
19.    }

```

**Kode Sumber 4.50 Fungsi Jalan 6**



```

1.     public void jalan7()
2.     {
3.         counter7++;
4.         if (counter7 % 2 == 0)
5.         {
6.             mybutton7.image.overrideSprite = jalan;
7.
8.             PlayerPrefs.SetInt("2jalan7", 0);
9.             Debug.Log(PlayerPrefs.GetInt("2jalan7"));
10.        };
11.        }
12.        else
13.        {
14.            mybutton7.image.overrideSprite = pencet
15.            ;
16.            PlayerPrefs.SetInt("2jalan7", 20);
17.            Debug.Log(PlayerPrefs.GetInt("2jalan7"));
18.        };
19.    }
20. }

```

#### Kode Sumber 4.51 Fungsi Jalan 7

```

1.     public void jalan8()
2.     {
3.         counter8++;
4.         if (counter8 % 2 == 0)
5.         {
6.             mybutton8.image.overrideSprite = jalan;
7.
8.             PlayerPrefs.SetInt("2jalan8", 0);
9.             Debug.Log(PlayerPrefs.GetInt("2jalan8"));
10.        };
11.        }
12.        else
13.        {
14.            mybutton8.image.overrideSprite = pencet
15.            ;
16.            PlayerPrefs.SetInt("2jalan8", 20);
17.            Debug.Log(PlayerPrefs.GetInt("2jalan8"));
18.        };
19.    }
20. }

```

```

15.         }
16.     }

```

#### Kode Sumber 4.52 Fungsi Jalan 8

```

1.     public void jalan9()
2.     {
3.
4.         counter9++;
5.         if (counter9 % 2 == 0)
6.         {
7.             mybutton9.image.overrideSprite = jalan;
8.
9.             PlayerPrefs.SetInt("2jalan9", 0);
10.            Debug.Log(PlayerPrefs.GetInt("2jalan9"))
11.        );
12.    }
13.    else
14.    {
15.        mybutton9.image.overrideSprite = pencet
16.    ;
17.        PlayerPrefs.SetInt("2jalan9", 20);
18.        Debug.Log(PlayerPrefs.GetInt("2jalan9"))
19.    );
20.    }

```

#### Kode Sumber 4.53 Fungsi Jalan 9

### 4.3.5.4 Implementasi Pembuatan Soal 4

Soal empat mengimplementasikan graf sederhana yaitu menghitung jumlah jarak. Terdapat 3 pilihan jawaban bergambar satuan kilometer, jika peserta memilih satuan kilometer yang sesuai dengan pertanyaan maka jawaban peserta tersebut adalah benar. Inisiasi variabel satuan kilometer dapat dilihat pada Kode Sumber 4.54. Fungsi untuk pilihan satuan kilometer yang benar dapat dilihat pada Kode Sumber 4.57, fungsi untuk pilihan satuan kilometer yang salah dapat dilihat pada Kode Sumber 4.55 dan Kode Sumber 4.56.

```

1.  public Button mybutton1, mybutton2, mybutton3;
2.      public Sprite km4putih;
3.      public Sprite km5putih;
4.      public Sprite km9putih;
5.
6.      public Sprite km4kuning;
7.      public Sprite km5kuning;
8.      public Sprite km9kuning;
9.
10.     private int counter1, counter2, counter3 = 0;

```

#### Kode Sumber 4.54 Inisiasi Variabel Satuan Kilometer

```

1.  public void tmb11()
2.  {
3.      counter1++;
4.      if (counter1 % 2 == 0)
5.      {
6.          mybutton1.image.overrideSprite = km4put
7.      ih;
8.          PlayerPrefs.SetInt("km1", 0);
9.          Debug.Log(PlayerPrefs.GetInt("km1"));
10.     } else
11.     {
12.         mybutton1.image.overrideSprite = km4kun
13.     ing;
14.         PlayerPrefs.SetInt("km1", 0);
15.         Debug.Log (PlayerPrefs.GetInt ("km1"));
16.     }
17. }

```

#### Kode Sumber 4.55 Fungsi Tombol 1

```

1.  public void tmb12()
2.  {
3.      counter2++;
4.      if (counter2 % 2 == 0)
5.      {
6.          mybutton2.image.overrideSprite = km5put
7.      ih;
8.          PlayerPrefs.SetInt("km2", 0);

```

```

8.         Debug.Log(PlayerPrefs.GetInt("km2"));
9.     } else
10.    {
11.        mybutton2.image.overrideSprite = km5kun
12.        ing;
13.        PlayerPrefs.SetInt("km2", 0);
14.        Debug.Log (PlayerPrefs.GetInt ("km2"));
15.    }
16.

```

**Kode Sumber 4.56 Fungsi Tombol 2**

```

1.  public void tmb13()
2.  {
3.      counter3++;
4.      if (counter3 % 2 == 0)
5.      {
6.          mybutton3.image.overrideSprite = km9put
7.          ih;
8.          PlayerPrefs.SetInt("km3", 0);
9.          Debug.Log(PlayerPrefs.GetInt("km3"));
10.     } else
11.     {
12.         mybutton3.image.overrideSprite = km9kun
13.         ing;
14.         PlayerPrefs.SetInt("km3", 1);
15.         Debug.Log (PlayerPrefs.GetInt ("km3"));
16.     }
17. }

```

**Kode Sumber 4.57 Fungsi Tombol 3**

#### **4.3.5.5 Implementasi Pembuatan Pengurutan Nomor Soal Secara Acak**

Soal-soal yang diberikan memiliki urutan nomor secara acak, sehingga satu peserta dengan peserta yang lain memiliki kemungkinan untuk mengerjakan urutan nomor yang berbeda.

Fungsi acak pada urutan nomor dapat dilihat pada Kode Sumber 4.58.

```

1.      int scene;
2.      int min = 0;
3.      int max = 3;
4.      int min1 = 0;
5.      int max1 = 2;
6.      int[] number = new int [3] {1,2,3};
7.      int[] number1 = new int [2] {2,3};
8.      int[] number2 = new int [2] {1,3};
9.      int[] number3 = new int [2] {1,2};

```

**Kode Sumber 4.58 Inisiasi Variabel Jumlah Halaman Soal**

#### 4.3.5.6 Implementasi *Random* Soal dan *Insert* Nilai

Pada saat peserta mengerjakan soal, sistem akan mengacak soal. Terdapat empat soal yang akan diacak oleh sistem. Setiap peserta selesai mengerjakan soal, maka sistem akan memasukkan nilai peserta pada soal tersebut ke dalam *database*. Fungsi *random* soal dan *insert* nilai dapat dilihat pada Kode Sumber 4.59 dan 4.60, fungsi kontrol *insert* nilai dapat dilihat pada Kode Sumber 4.61.

```

1.  if (jawaban.text == "Benar")
2.      {
3.          PlayerPrefs.SetInt("scoreno1", 25);
4.      } else {
5.          PlayerPrefs.SetInt("scoreno1", -5);
6.      }
7.      int cekno1 = PlayerPrefs.GetInt ("scoreno1"
8.      );
9.      int cekno2 = PlayerPrefs.GetInt ("scoreno2"
10.     );
11.     int cekno0 = PlayerPrefs.GetInt ("scoreno0"
12.     );
13.     int cekno3 = PlayerPrefs.GetInt ("scoreno3"
14.     );
15.     score1 = PlayerPrefs.GetInt("scoreno1");

```

```

12.         Debug.Log (PlayerPrefs.GetInt ("scoreno1"))
13.     ;
14.     user = PlayerPrefs.GetString("username").ToString();
15.     StartCoroutine(Postscore(user,idsoal, score
16.     1, 5f));
17.
18.     string acak = scene.ToString();
19.     if (cekno1 != 0 && cekno2 != 0 && cekno0 !=
20.     0 && cekno3 != 0) {
21.         Application.LoadLevel ("pageSelesai");
22.     }
23.     else if (cekno1 != 0 && cekno2 == 0 && cekn
24.     o0 == 0 && cekno3 == 0) {
25.         scene = number [UnityEngine.Random.Rang
26.         e (min, max)];
27.         acak = scene.ToString ();
28.         Debug.Log (acak);
29.         Application.LoadLevel (acak);
30.     } else if (cekno1 != 0 && cekno2 == 0 && ce
31.     kno0 != 0 && cekno3 == 0) {
32.         scene = number1 [UnityEngine.Random.Ran
33.         ge (min1, max1)];
34.         print (scene);
35.         acak = scene.ToString ();
36.         Debug.Log (acak);
37.         Application.LoadLevel (acak);
38.     } else if (cekno1 != 0 && cekno2 != 0 && ce
39.     kno0 == 0 && cekno3 == 0) {
40.         scene = number2 [UnityEngine.Random.Ran
41.         ge (min1, max1)];
42.         print (scene);
43.         acak = scene.ToString ();
44.         Debug.Log (acak);
45.         Application.LoadLevel (acak);

```

```

43.         } else if (cekno0 == 0 && cekno2 != 0 && ce
kno1 != 0 && cekno3 != 0 ) {
44.             Application.LoadLevel ("0");
45.         } else if (cekno0 != 0 && cekno2 == 0 && ce
kno1 != 0 && cekno3 != 0) {
46.             Application.LoadLevel ("2");
47.         } else if (cekno0 != 0 && cekno2 != 0 && ce
kno1 != 0 && cekno3 == 0) {
48.             Application.LoadLevel ("3");
49.         }

```

### Kode Sumber 4.59 Implementasi *Random Soal*

```

1. IEnumerator Postscore(string user, string idsoal, i
nt nilai1, float delay)
2.     {
3.         UnityWebRequest link = UnityWebRequest.Get
("http://somethingnotright.dx.am/masukinnilai.php?r
_username=" + user + "&r_idsoal=" + idsoal + "&r_ni
lai=" + nilai1);
4.         yield return link.Send ();
5.         yield return new WaitForSeconds(delay);
6.         if (link.isError) {
7.             Debug.Log (link.error);
8.         }
9.     }

```

### Kode Sumber 4.60 Implementasi *Insert Nilai*

```

1. <?php
2. header("Access-Control-Allow-Credentials: true");
3. header('Access-Control-Allow-Origin: *');
4. header('Access-Control-Allow-
Methods: POST, GET, OPTIONS');
5. header('Access-Control-Allow-Headers: Accept, X-
Access-Token, X-Application-Name, X-Request-Sent-
Time');
6.
7. error_reporting(E_ALL ^ E_DEPRECATED);
8. $Server = "fdb17.awardspace.net";
9. $User = "2326813_admin";

```

```

10. $Pass = "zzz1312000";
11. $Db = "2326813_admin";
12. $Connection = mysql_connect($Server, $User, $Pass,
    $Db);
13.
14. if(mysql_error()){
15.     die("Connection Failed!=". mysql_error());
16. }
17. mysql_select_db("2326813_admin", $Connection) or die("Could not load to database" . mysql_error());
18.
19. $username = $_GET["r_username"];
20. $nilai = $_GET["r_nilai"];
21. $idsoal = $_GET["r_idsoal"];
22. $check = mysql_query("SELECT * FROM peserta WHERE `
    username` = '". $username.'");
23. $numrows = mysql_num_rows($check);
24. if($numrows == 0)
25. {
26.     die("Error");
27. }
28. else
29. {
30.     while($row = mysql_fetch_assoc($check))
31.     {
32.         if($username == $row['username'])
33.         {
34.             $sql = mysql_query("UPDATE jawab INNER
                JOIN peserta ON jawab.id=peserta.id SET jawab.skor
                = '". $nilai.'" WHERE peserta.username = '". $usernam
                e.'" and jawab.id_soal = '". $idsoal.'");
35.             die("berhasil");
36.         }
37.         else
38.         {
39.             die("error");
40.         }
41.     }
42. }
43. ?>

```

**Kode Sumber 4.61 Kontrol *Insert* Nilai**



#### 4.3.5.7 Implementasi Waktu Hitung Mundur Soal

Pada saat peserta mengerjakan soal, terdapat batas waktu untuk mengerjakan pada soal tersebut. Setiap soal mempunyai batas waktu yang berbeda-beda tergantung tingkat kesulitan soal. Fungsi waktu hitung mundur dapat dilihat pada Kode Sumber 4.62.

```

1. public float StartTimes, t;
2. public string minutes, seconds;
3. StartTimes = 300;
4. t = StartTimes - Time.timeSinceLevelLoad;
5. minutes = ((int)t / 60).ToString();
6. seconds = (t % 60).ToString("f0");
7. timerText.text = minutes + " : " + seconds;
8. if (t <= 0) {
9.     timerText.text = "Waktu Habis";
10.    timerText.color = Color.red;
11.    PlayerPrefs.SetInt("scoreno1", -5);
12.    int cekno1 = PlayerPrefs.GetInt ("score
no1");
13.    int cekno2 = PlayerPrefs.GetInt ("score
no2");
14.    int cekno0 = PlayerPrefs.GetInt ("score
no0");
15.    int cekno3 = PlayerPrefs.GetInt ("score
no3");
16.    score1 = PlayerPrefs.GetInt("scoreno1")
;
17.    Debug.Log (PlayerPrefs.GetInt ("scoreno
1"));
18.    user = PlayerPrefs.GetString("username"
).ToString();
19.    StartCoroutine(Postscore(user,idsoal, s
core1, 5f));
20.    string acak = scene.ToString();
21.    if (cekno1 != 0 && cekno2 != 0 && cekno
0 != 0 && cekno3 != 0) {
22.        Application.LoadLevel ("pageSelesai
"); }
23.    else if (cekno1 != 0 && cekno2 == 0 &&
cekno0 == 0 && cekno3 == 0) {

```

```

24.         scene = number [UnityEngine.Random.
    Range (min, max)];
25.         acak = scene.ToString ();
26.         Debug.Log (acak);
27.         Application.LoadLevel (acak);
28.     } else if (cekno1 != 0 && cekno2 == 0 &
    & cekno0 != 0 && cekno3 == 0) {
29.         scene = number1 [UnityEngine.Random
    .Range (min1, max1)];
30.         print (scene);
31.         acak = scene.ToString ();
32.         Debug.Log (acak);
33.         Application.LoadLevel (acak);
34.     } else if (cekno1 != 0 && cekno2 != 0 &
    & cekno0 == 0 && cekno3 == 0) {
35.         scene = number2 [UnityEngine.Random
    .Range (min1, max1)];
36.         print (scene);
37.         acak = scene.ToString ();
38.         Debug.Log (acak);
39.         Application.LoadLevel (acak);
40.     } else if (cekno1 != 0 && cekno2 == 0 &
    & cekno0 == 0 && cekno3 != 0) {
41.         scene = number3 [UnityEngine.Random
    .Range (min1, max1)];
42.         print (scene);
43.         acak = scene.ToString ();
44.         Debug.Log (acak);
45.         Application.LoadLevel (acak);
46.     } else if (cekno0 == 0 && cekno2 != 0 &
    & cekno1 != 0 && cekno3 != 0 ) {
47.         Application.LoadLevel ("0");
48.     } else if (cekno0 != 0 && cekno2 == 0 &
    & cekno1 != 0 && cekno3 != 0) {
49.         Application.LoadLevel ("2");
50.     } else if (cekno0 != 0 && cekno2 != 0 &
    & cekno1 != 0 && cekno3 == 0) {
51.         Application.LoadLevel ("3");
52.     }

```

**Kode Sumber 4.62 Implementasi Waktu Hitung Mundur**

Pada kode di atas, dijelaskan bahwa jika peserta tidak dapat menjawab hingga batas waktu yang diberikan pada masing-masing soal, maka jawaban peserta pada soal tersebut akan dianggap salah.

#### 4.3.5.8 Implementasi Terjemahan Bahasa

Terdapat dua bahasa yang tersedia ketika mengerjakan soal yaitu bahasa Indonesia dan bahasa Inggris, peserta dapat memilih salah satu bahasa yang telah tersedia tersebut. Fungsi terjemahan bahasa dapat dilihat pada Kode Sumber 4.63.

```

1. public class OCL : MonoBehaviour
2. {
3.     public delegate void ActiveChanged(bool isActive);
4.     public static ActiveChanged onActiveChanged;
5.     public delegate void LanguageChanged(System
6.         Language oldLang, SystemLanguage newLang);
7.     public static LanguageChanged onLanguageChanged;
8.     public delegate void LanguagesChanged();
9.     public static LanguagesChanged onLanguagesChanged;
10.    public delegate void LocalizationChanged(object id, SystemLanguage language, object newValue);
11.    public static LocalizationChanged onLocalizationChanged;
12.    protected static LocalizationSetup runtimeSetup;
13.    protected static string parameterPattern = @"\";
14.    protected static List<string> stringsWithParameters;
15.    protected static List<Regex> stringsWithParametersRegex;
16.    static OCL()
17.    {

```

```

17.         LocalizationSetup persistedSetup = Resources.Load<LocalizationSetup>(LocalizationSetup.setupResourceName);
18.         if (persistedSetup == null)
19.         {
20.             Debug.LogWarning("[OCL] No setup as set <" + LocalizationSetup.setupResourceName + "> found on disk. Initializing deactivated empty setup.");
21.             runtimeSetup = ScriptableObject.CreateInstance<LocalizationSetup>();
22.             runtimeSetup.active = false;
23.             runtimeSetup.AddLanguage(runtimeSetup.defaultLanguage);
24.         }
25.         else
26.         {
27.             runtimeSetup = (LocalizationSetup)persistedSetup.Clone();
28.             runtimeSetup.LoadSubAssets(true);
29.         }
30.         initStringsWithParameters();
31.     }
32.     protected static void initStringsWithParameters()
33.     {
34.         stringsWithParameters = new List<string>();
35.         stringsWithParametersRegex = new List<Regex>();
36.
37.         foreach(string stringId in runtimeSetup.GetIds<string>())
38.         {
39.             string[] substrings = Regex.Split(stringId, parameterPattern);
40.             if (substrings.Length > 1)
41.             {
42.                 int index = 0;
43.                 string res = "";
44.                 foreach (string match in substrings) {
45.                     if (index % 2 == 0)

```

```

46.         {
47.             res += match;
48.         }
49.         else
50.         {
51.             res += "(.*)";
52.         }
53.         index++;
54.     }
55.     Regex stringRegex = new Regex(r
es, RegexOptions.IgnoreCase);
56.     stringsWithParameters.Add(strin
gId);
57.     stringsWithParametersRegex.Add(
stringRegex); }
58.     }
59.     } public static bool IsActive()
60.     {
61.         return runtimeSetup.active;
62.     }
63.     /// <param name="value"></param>
64.     public static void SetActive(bool value)
65.     {
66.         runtimeSetup.active = value;
67.         if(onActiveChanged != null)
68.         {
69.             onActiveChanged(IsActive());
70.         }
71.     }
72.     public static SystemLanguage GetLanguage()
73.     {
74.         if (IsLanguageAuto())
75.         {
76.             return Application.systemLanguage;
77.         }
78.         else
79.         {
80.             return runtimeSetup.forcedLanguage;
            }
        }

```

**Kode Sumber 4.63 Implementasi Terjemahan Bahasa**

*[Halaman ini sengaja dikosongkan]*

## **BAB V**

### **PENGUJIAN DAN EVALUASI**

Bab ini membahas tentang pengujian dan evaluasi pada perangkat lunak yang dibangun untuk tugas akhir ini. Pengujian dilakukan pada kasus penggunaan dari sistem perangkat lunak.

#### **5.1 Lingkungan Uji Coba**

Pada proses pengujian perangkat lunak, dibutuhkan suatu lingkungan pengujian yang sesuai dengan standar kebutuhan. Lingkungan pengujian dalam tugas akhir ini dilakukan pada setiap kasus penggunaan. Spesifikasi masing-masing lingkungan pengujian dijabarkan pada Tabel 5.1.

**Tabel 5.1 Lingkungan Pengujian Fungsionalitas Perangkat Lunak**

No	Spesifikasi	Deskripsi
1	Jenis Perangkat	Laptop
	Merek Perangkat	Vaio
	Sistem Operasi	Sistem operasi Windows 7 (64-bit)
	CPU	Processor Intel(R) Core(TM) i7-2640M CPU @ 2.80GHz
	RAM	4 GB
2	Jenis Perangkat	Galaxy Tab A
	Merek Perangkat	Samsung
	Sistem Operasi	Android 6.0.1Marshmallow
	CPU	8-core
	RAM	2 GB

#### **5.2 Pengujian Fungsionalitas**

Pengujian fungsionalitas aplikasi ini dapat dilakukan secara mandiri. Pengujian ini bertujuan untuk mengetahui kesesuaian

keluaran dari tiap tahap dan langkah penggunaan fitur terhadap skenario yang dipersiapkan. Skenario yang dibuat mengacu pada kasus penggunaan yang telah dijelaskan pada subbab 3.1.3.4. Skenario uji coba fungsionalitas yang dilakukan terhadap aplikasi yang dibangun dijelaskan pada Tabel 5.2.

**Tabel 5.2 Skenario Uji Coba Fungsionalitas**

<b>Kode Uji Coba</b>	<b>Nama Uji Coba</b>
UC-0001	Uji melihat tentang Bebras
UC-0002	Uji menambah detail peserta
UC-0003	Uji mengerjakan soal
UC-0004	Uji melihat nilai
UC-0005	Uji melihat peringkat

### **5.2.1 Pengujian Fungsionalitas Melihat Tentang Bebras**

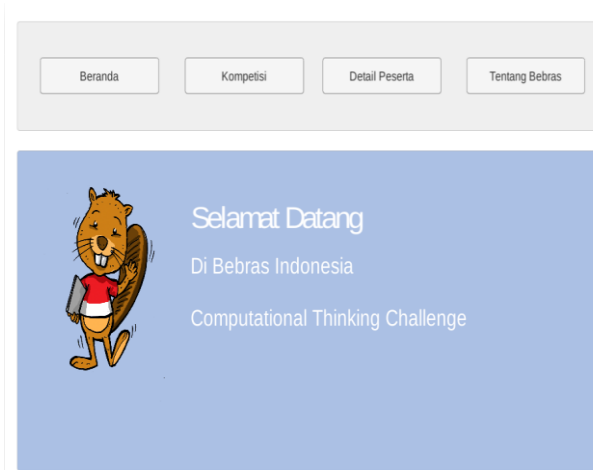
**Tabel 5.3 Uji Coba Melihat Tentang Bebras**

<b>No.Pengujian</b>	<b>UJ-UC-0001</b>
<b>Skenario Pengujian</b>	Aplikasi menampilkan informasi tentang Bebras ketika masuk ke halaman Tentang Bebras
<b>Kondisi Awal</b>	Aplikasi menampilkan halaman Beranda yang terdapat beberapa menu, yaitu “Beranda”, “Kompetisi”, “Detail Peserta”, “Tentang Bebras”
<b>Aksi yang dilakukan</b>	Memilih tombol “Tentang Bebras”
<b>Hasil yang diharapkan</b>	Aplikasi dapat menampilkan informasi tentang Bebras
<b>Hasil yang diperoleh</b>	Aplikasi menampilkan informasi tentang Bebras
<b>Hasil Pengujian</b>	Berhasil

Langkah-langkah UJ-UC-0001 berdasarkan pada Tabel 5.3:

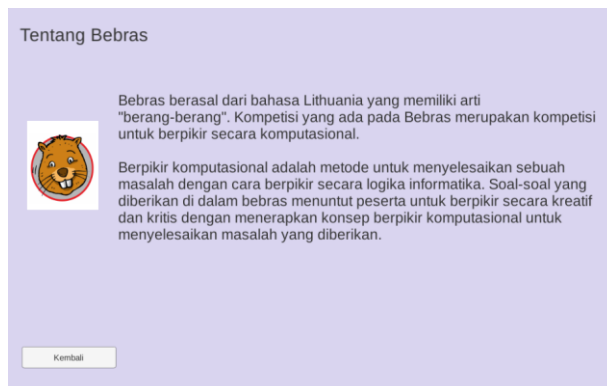
1. Peserta masuk ke halaman Beranda seperti pada Gambar 5.1





**Gambar 5.1 Halaman Beranda**

2. Peserta menekan tombol yang bertuliskan “Tentang Bebras”.
3. Sistem menampilkan informasi tentang Bebras seperti pada Gambar 5.2 yang menunjukkan bahwa peserta telah sukses melihat informasi tentang Bebras.



**Gambar 5.2 Halaman Tentang Bebras**

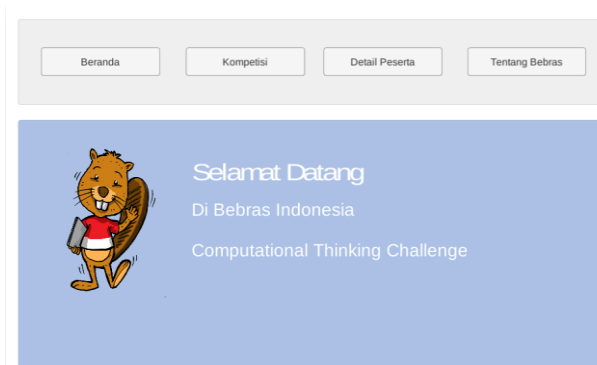
### 5.2.2 Pengujian Fungsionalitas Melihat Detail Peserta

**Tabel 5.4 Uji Coba Melihat Detail Peserta**

<b>No.Pengujian</b>	<b>UJ-UC-0002</b>
<b>Skenario Pengujian</b>	Aplikasi menampilkan data peserta ketika masuk ke halaman Detail Peserta
<b>Kondisi Awal</b>	Aplikasi menampilkan halaman Beranda yang terdapat beberapa menu, yaitu “Beranda”, “Kompetisi”, “Detail Peserta”, “Tentang Bebras”
<b>Aksi yang dilakukan</b>	Memilih tombol “Detail Peserta”
<b>Hasil yang diharapkan</b>	Aplikasi dapat menampilkan data peserta
<b>Hasil yang diperoleh</b>	Aplikasi menampilkan data peserta
<b>Hasil Pengujian</b>	Berhasil

Langkah-langkah UJ-UC-0002 berdasarkan pada Tabel 5.4:

1. Peserta masuk ke halaman Beranda seperti pada Gambar 5.3.



**Gambar 5.3 Halaman Beranda**

2. Peserta menekan tombol yang bertuliskan “Detail Peserta”.
3. Sistem menampilkan data peserta seperti pada Gambar 5.4 yang menunjukkan bahwa peserta telah sukses melihat data peserta.



Gambar 5.4 Halaman Detail Peserta

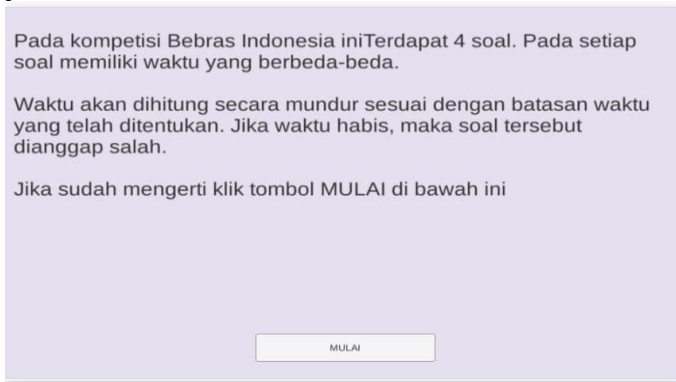
### 5.2.3 Pengujian Fungsionalitas Mengerjakan Soal

Tabel 5.5 Uji Coba Mengerjakan Soal

No.Pengujian	UJ-UC-0003
<b>Skenario Pengujian</b>	Peserta dapat mengerjakan soal pada sistem
<b>Kondisi Awal</b>	Peserta sudah <i>login</i> kemudian masuk ke halaman Penjelasan Soal yang terdapat menu “Mulai”
<b>Aksi yang dilakukan</b>	Memilih tombol “Mulai”
<b>Hasil yang diharapkan</b>	Peserta dapat mengerjakan soal pada sistem
<b>Hasil yang diperoleh</b>	Peserta mengerjakan soal pada sistem
<b>Hasil Pengujian</b>	Berhasil

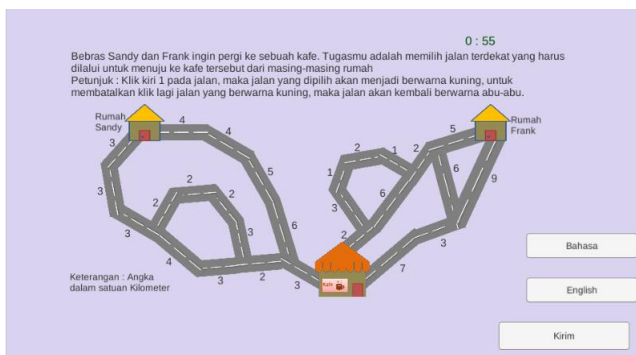
Langkah-langkah UJ-UC-0003 berdasarkan pada Tabel 5.5:

1. Peserta masuk ke halaman Penjelasan Soal.
2. Peserta menekan tombol yang bertuliskan “Mulai” seperti pada Gambar 5.5.



**Gambar 5.5 Halaman Penjelasan Soal**

3. Sistem menampilkan soal dan pilihan jawaban seperti pada Gambar 5.6, Gambar 5.7, Gambar 5.8, dan Gambar 5.9 yang menunjukkan bahwa peserta telah sukses melihat soal.



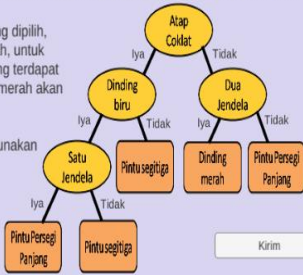

**Gambar 5.6 Halaman Soal 1**

0 : 54

Bebras Sandy ingin membuat rumah. Namun untuk membuat rumah ada aturannya menggunakan graf yang disebut dengan "pohon" karena ada simpul yang saling berhubungan. Pada setiap simpul anda harus memutuskan arah mana yang ingin anda pilih. Anda tidak dapat naik lagi.

Petunjuk : Klik kiri pada rumah yang dipilih, maka akan muncul lingkaran merah, untuk membatalkan, klik pada rumah yang terdapat lingkaran merah, maka lingkaran merah akan menghilang.

Rumah mana yang TIDAK menggunakan aturan yang benar?



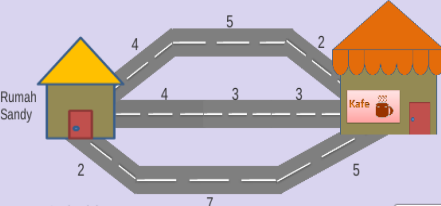
Kirim

Gambar 5.7 Halaman Soal 2

1 : 26

Bebras Sandy ingin pergi ke sebuah kafe. Tugasmu adalah memilih jalan terdekat yang harus dilalui untuk menuju ke kafe tersebut.

Petunjuk : Klik kiri pada jalan, maka jalan yang dipilih akan menjadi berwarna kuning, untuk membatalkan klik lagi jalan yang berwarna kuning, maka jalan akan kembali berwarna abu-abu.



Kirim

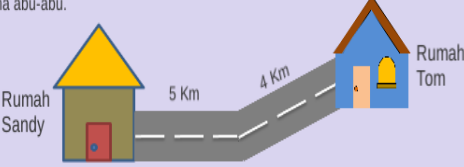
Keterangan : Angka dalam satuan Kilometer

Gambar 5.8 Halaman Soal 3

0 : 55

Bebas Sandy ingin pergi bermain ke rumah Bebas Tom. Berapakah jarak yang harus ditempuh dari rumah Bebas Sandy menuju rumah Bebas Tom?

Petunjuk : Klik kiri pada tombol yang dipilih, maka tombol yang dipilih akan menjadi berwarna kuning, untuk membatalkan klik lagi tombol yang berwarna kuning, maka tombol akan kembali berwarna abu-abu.



Pilihlah jumlah jarak yang benar dari rumah Bebas Sandy menuju rumah Bebas Tom

4 Km    5 Km    9 Km    Kirim

**Gambar 5.9 Halaman Soal 4**

4. Peserta dapat mengerjakan soal sesuai dengan petunjuk yang telah diberikan. Gambar 5.10, Gambar 5.11, Gambar 5.12, dan Gambar 5.13 adalah contoh soal yang telah dijawab.

0 : 43

Bebas Sandy dan Frank ingin pergi ke sebuah kafe. Tugasmu adalah memilih jalan terdekat yang harus dilalui untuk menuju ke kafe tersebut dari masing-masing rumah

Petunjuk : Klik kiri 1 pada jalan, maka jalan yang dipilih akan menjadi berwarna kuning, untuk membatalkan klik lagi jalan yang berwarna kuning, maka jalan akan kembali berwarna abu-abu.



Keterangan : Angka dalam satuan Kilometer

Bahasa    English    Kirim

**Gambar 5.10 Halaman Soal 1 yang Telah Dijawab**

0 : 44

Bebras Sandy ingin membuat rumah. Namun untuk membuat rumah ada aturannya menggunakan graf yang disebut dengan "pohon" karena ada simpul yang saling berhubungan. Pada setiap simpul anda harus memutuskan arah mana yang ingin anda pilih. Anda tidak dapat naik lagi.

Petunjuk : Klik kiri pada rumah yang dipilih, maka akan muncul lingkaran merah, untuk membatalkan, klik pada rumah yang terdapat lingkaran merah, maka lingkaran merah akan menghilang.

Rumah mana yang TIDAK menggunakan aturan yang benar?



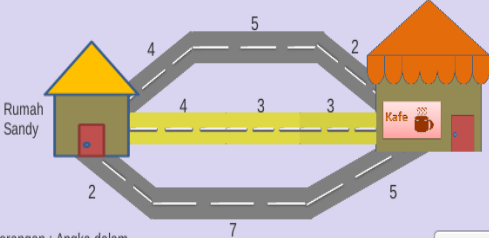
Kirim

Gambar 5.11 Halaman Soal 2 yang Telah Dijawab

1 : 21

Bebras Sandy ingin pergi ke sebuah kafe. Tugasmu adalah memilih jalan terdekat yang harus dilalui untuk menuju ke kafe tersebut.

Petunjuk : Klik kiri pada jalan, maka jalan yang dipilih akan menjadi berwarna kuning, untuk membatalkan klik lagi jalan yang berwarna kuning, maka jalan akan kembali berwarna abu-abu.



Kirim

Keterangan : Angka dalam satuan Kilometer

Gambar 5.12 Halaman Soal 3 yang Telah Dijawab

0 : 36

Bebras Sandy ingin pergi bermain ke rumah Bebras Tom. Berapakah jarak yang harus ditempuh dari rumah Bebras Sandy menuju rumah Bebras Tom?

Petunjuk : Klik kiri pada tombol yang dipilih, maka tombol yang dipilih akan menjadi berwarna kuning, untuk membatalkan klik lagi tombol yang berwarna kuning, maka tombol akan kembali berwarna abu-abu.

Pilihlah jumlah jarak yang benar dari rumah Bebras Sandy menuju rumah Bebras Tom

**Gambar 5.13 Halaman Soal 4 yang Telah Dijawab**

5. Peserta yang telah selesai mengerjakan soal dapat menyimpan nilai dan melanjutkan ke soal berikutnya dengan memilih tombol “Kirim”.

#### 5.2.4 Pengujian Fungsionalitas Melihat Nilai

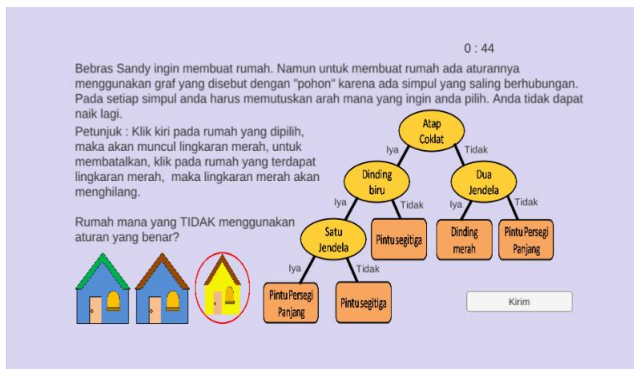
**Tabel 5.6 Uji Coba Melihat Nilai**

<b>No.Pengujian</b>	<b>UJ-UC-0004</b>
<b>Skenario Pengujian</b>	Aplikasi menampilkan data nilai ketika masuk ke halaman Selesai
<b>Kondisi Awal</b>	Aplikasi menampilkan halaman Soal yang terdapat menu yaitu “Kirim”
<b>Aksi yang dilakukan</b>	Memilih tombol “Kirim”
<b>Hasil yang diharapkan</b>	Aplikasi dapat menampilkan data nilai yang didapat oleh peserta
<b>Hasil yang diperoleh</b>	Aplikasi menampilkan data nilai yang didapat oleh peserta
<b>Hasil Pengujian</b>	Berhasil



Langkah-langkah UJ-UC-0004 berdasarkan pada Tabel 5.6:

1. Peserta masuk ke halaman Soal
2. Peserta menekan tombol yang bertuliskan “Kirim” seperti pada Gambar 5.14.



**Gambar 5.14 Halaman Soal**

3. Jika semua soal telah dikerjakan, maka sistem akan menampilkan data nilai yang didapat oleh peserta yang menunjukkan bahwa peserta telah sukses melihat data nilai seperti pada Gambar 5.15.



**Gambar 5.15 Halaman Selesai**

### 5.2.5 Pengujian Fungsionalitas Melihat Peringkat

**Tabel 5.7 Uji Coba Melihat Peringkat**

No.Pengujian	UJ-UC-0005
<b>Skenario Pengujian</b>	Aplikasi menampilkan data peringkat ketika masuk ke halaman Peringkat
<b>Kondisi Awal</b>	Aplikasi menampilkan halaman Selesai yang terdapat beberapa menu, yaitu “RANKING” dan “Log Out”
<b>Aksi yang dilakukan</b>	Memilih tombol “RANKING”
<b>Hasil yang diharapkan</b>	Aplikasi dapat menampilkan data peringkat berupa data semua nilai peserta
<b>Hasil yang diperoleh</b>	Aplikasi menampilkan data peringkat berupa data semua nilai peserta
<b>Hasil Pengujian</b>	Berhasil

Langkah-langkah UJ-UC-0005 berdasarkan pada Tabel 5.7:

1. Peserta masuk ke halaman Selesai
2. Peserta menekan tombol yang bertuliskan “RANKING” seperti pada Gambar 5.16.



**Gambar 5.16 Halaman Selesai**

3. Sistem menampilkan data peringkat berupa data semua nilai peserta seperti pada Gambar 5.17 yang menunjukkan bahwa peserta telah sukses melihat peringkat



**Gambar 5.17 Halaman Peringkat**

### 5.3 Evaluasi Pengujian

Pada subbab ini akan diberikan hasil evaluasi dari pengujian-pengujian yang telah dilakukan. Evaluasi yang diberikan meliputi evaluasi pengujian fungsional.

Evaluasi pengujian fungsionalitas dilakukan dengan menampilkan data rekapitulasi aplikasi yang telah dipaparkan pada subbab 5.2, dalam hal ini rekapitulasi disusun dalam bentuk tabel yang dapat dilihat pada Tabel 5.8. Dari data yang terdapat pada tabel tersebut, diketahui bahwa aplikasi yang dibuat telah memenuhi kasus penggunaan yang telah ditentukan.

**Tabel 5.8 Rekapitulasi Hasil Uji Fungsionalitas**

Kode	Uji Coba	Hasil
UJ-UC-0001	Uji melihat tentang Bebras	Berhasil
UJ-UC-0002	Uji menambah detail peserta	Berhasil
UJ-UC-0003	Uji mengerjakan soal	Berhasil
UJ-UC-0004	Uji melihat nilai	Berhasil
UJ-UC-0005	Uji melihat peringkat	Berhasil

*[Halaman ini sengaja dikosongkan]*

## **BAB VI**

### **KESIMPULAN DAN SARAN**

Bab ini membahas mengenai kesimpulan yang didapatkan dari uji tujuan pembuatan perangkat lunak dan hasil uji coba yang telah dilakukan. Selain kesimpulan, terdapat pula saran-saran untuk pengembangan perangkat lunak lebih lanjut.

#### **6.1. Kesimpulan**

Dari proses pengerjaan tugas akhir selama tahap analisis, desain, implementasi, hingga pengujian didapatkan kesimpulan sebagai berikut:

1. Berdasarkan hasil uji coba fungsionalitas, aplikasi berhasil dibangun sesuai rancangan.
2. Berdasarkan kuesioner, responden setuju antarmuka aplikasi nyaman dengan nilai 3,9 dari nilai maksimal 5.
3. Berdasarkan kuesioner, responden sangat setuju performa aplikasi baik dengan nilai 4,15 dari nilai maksimal 5.
4. Berdasarkan kuesioner, responden setuju soal-soal yang diberikan menarik dengan nilai 4,65 dari nilai maksimal 5.

#### **6.2. Saran**

Berikut merupakan beberapa saran untuk perbaikan dan pengembangan aplikasi dimasa yang akan datang:

1. Gambar-gambar yang dipakai pada soal dibuat semenarik mungkin agar pengguna tertarik.
2. Menambah fitur panduan mengerjakan soal.

*[Halaman ini sengaja dikosongkan]*

## DAFTAR PUSTAKA

- [1] “Beranda | Bebras Indonesia.” [Daring]. Tersedia pada: <http://Bebras.or.id/>. [Diakses: 02-Jun-2017].
- [2] “What is Bebras | [www.Bebbras.org](http://www.Bebbras.org).” [Daring]. Tersedia pada: <http://Bebras.org/>. [Diakses: 02-Jun-2017].
- [3] “TOKI SELENGGARAKAN BEBRAS INDONESIA CHALLENGE 2016 » Tim Olimpiade Komputer Indonesia.” [Daring]. Tersedia pada: <http://www.toki.or.id/2016/toki-selenggarakan-Bebbras-indonesia-challenge-2016/>. [Diakses: 02-Jun-2017].
- [4] Dagienė, V., Stupurienė, G. Bebras – A Sustainable Community Building Model for the Concept Based Learning of Informatics and Computational Thinking. *Informatics in Education*, Vol. 15 (1), 25-44, 2016.
- [5] “Mengenal WebGL - CodePolitan.com.” [Daring]. Tersedia pada: <https://www.codepolitan.com/mengenal-webgl>. [Diakses: 02-Jun-2017].
- [6] “Teori Graf,” *Wikipedia*. 24-Nov-2016.
- [7] D. Jungnickel, *Graphs, Networks, and Algorithms* Fourth Edition, Berlin: Springer, 2013.
- [8] “Unity (game engine),” *Wikipedia*. 28-Jun-2017.
- [9] “C sharp,” *Wikipedia bahasa Indonesia, ensiklopedia bebas*. 13-Jun-2013.
- [10] B. Lawson and R. Sharp, *Introducing HTML5* Second Edition, Barkeley: New Riders, 2012.
- [11] L. Webster, “Visual Studio IDE, Code Editor, Team Services, & Mobile Center,” *Visual Studio*, 22-Jun-2017. [Daring]. Tersedia pada: <https://www.visualstudio.com/>. [Diakses: 11-Jul-2017].

*[Halaman ini sengaja dikosongkan]*



## **LAMPIRAN A**

### **PANDUAN PENGGUNAAN APLIKASI**

Berikut ini adalah lampiran panduan penggunaan aplikasi mulai dari pendaftaran hingga selesai mengerjakan soal.

#### **Mengakses Aplikasi**

Berikut ini akan dijelaskan cara pengguna mengakses aplikasi:

1. Buka aplikasi pada perangkat Android atau buka pada *browser*.
2. Pada halaman awal dari aplikasi terdapat empat menu yaitu Beranda, Kompetisi, Detail Peserta, dan Tentang Bebras seperti pada Gambar A.7.1.



**Gambar A.7.1 Halaman Beranda**

1. Tombol menuju halaman Beranda.
2. Tombol menuju halaman Kompetisi.
3. Tombol menuju halaman Detail Peserta.
4. Tombol menuju halaman Tentang Bebras.

## Mengakses Halaman Detail Peserta

Berikut ini akan dijelaskan cara pengguna mengakses halaman Kompetisi dan dijelaskan fitur pada halaman Detail Peserta:

1. Jika pada halaman Beranda pengguna memilih tombol Detail Peserta maka pengguna akan diarahkan menuju halaman Detail Peserta.
2. Pada halaman Detail Peserta terdapat informasi seperti jumlah peserta, detail peserta yang berisi *username*, nama dan asal provinsi peserta seperti pada Gambar A.7.2.



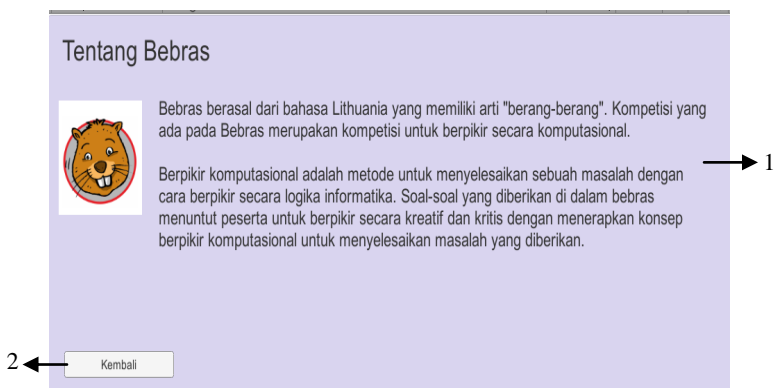
**Gambar A.7.2 Halaman Detail Peserta**

1. Teks untuk menampilkan jumlah peserta.
2. Teks untuk menampilkan detail peserta.
3. Teks untuk menampilkan jumlah peserta per provinsi.
4. Tombol Kembali ke halaman Beranda.

## Mengakses Halaman Tentang Bebras

Berikut ini akan dijelaskan cara pengguna mengakses halaman Tentang Bebras:

1. Jika pada halaman Beranda pengguna memilih tombol tentang Bebras maka pengguna akan diarahkan menuju halaman Tentang Bebras.
2. Pada halaman Tentang Bebras hanya berisi tulisan dan tombol kembali ke halaman Beranda seperti pada Gambar A.7.3.



**Gambar A.7.3 Halaman Tentang Bebras**

1. Teks tentang Bebras.
2. Tombol Kembali ke halaman Beranda.

### Mengakses Halaman Kompetisi

Berikut ini akan dijelaskan cara pengguna mengakses halaman Kompetisi:

1. Jika pada halaman Beranda pengguna memilih tombol Kompetisi maka pengguna akan diarahkan menuju halaman Kompetisi.
2. Pada halaman Kompetisi terdapat sebuah tombol untuk masuk halaman Login seperti pada Gambar A.7.4.



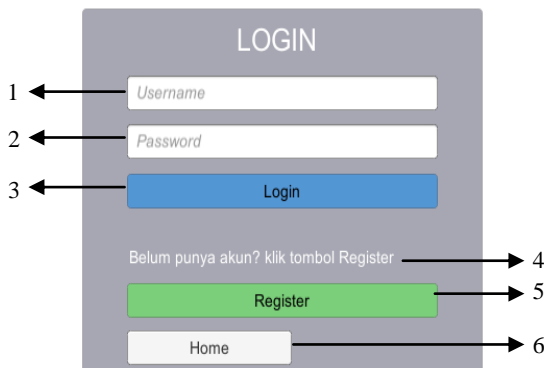
**Gambar A.7.4 Halaman Kompetisi**

1. Tombol untuk menuju halaman Beranda.
2. Tombol untuk menuju halaman Kompetisi.
3. Tombol untuk menuju halaman Tentang Bebras.
4. Tombol untuk menuju halaman Detail Peserta.
5. Tombol untuk menuju halaman Login.

### Mengakses Halaman Login

Berikut ini akan dijelaskan cara pengguna mengakses halaman Login dan dijelaskan fitur pada halaman Login:

1. Jika pada halaman Kompetisi pengguna memilih tombol Login maka pengguna akan diarahkan menuju halaman Login.
2. Pada halaman Login terdapat tombol Register jika pengguna belum memiliki akun untuk *login*. Dan terdapat dua *input fields* untuk memasukkan *username* dan *password* seperti pada Gambar A.7.5.



**Gambar A.7.5 Halaman Login**

1. *Input Field* untuk memasukkan *username*.
2. *Input Field* untuk memasukkan *password*.
3. Tombol untuk melakukan *login*.
4. Tombol untuk melakukan *register*.
5. Tombol untuk menuju halaman Beranda.

## Mengakses Halaman Register

Berikut ini akan dijelaskan cara pengguna mengakses halaman Register dan dijelaskan fitur pada halaman Register:

1. Saat pengguna berada pada halaman Login dan belum memiliki akun, maka pengguna diminta untuk membuat akun terlebih dahulu dengan cara memilih tombol Register pada halaman Login.
2. Saat pengguna berada pada halaman Register, pengguna diharuskan mengisi beberapa *field* untuk data peserta seperti yang dapat dilihat pada Gambar A.7.6.

The image shows a web form titled "REGISTER" on a light purple background. The form contains several input fields and buttons. Numbered callouts point to the following elements:

- 1: Points to the "Username" input field.
- 2: Points to the "Nama" input field.
- 3: Points to the "Password" input field.
- 4: Points to the "Ulangi Password" input field.
- 5: Points to the "Pilih Provinsi Anda" dropdown menu.
- 6: Points to the green "Register" button.
- 7: Points to the "Kembali" button.

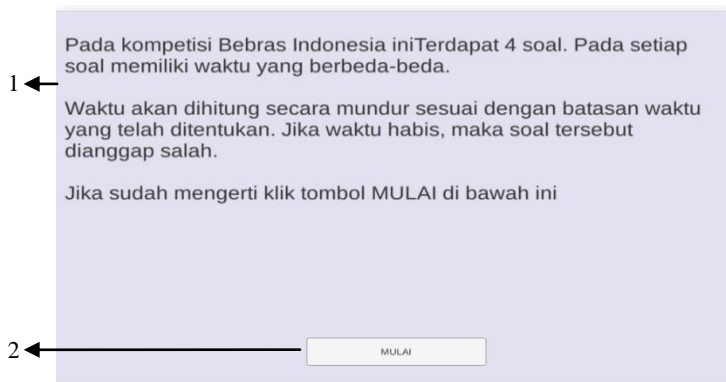
**Gambar A.7.6 Halaman Register**

1. *Input Field* untuk *Username*.
2. *Input Field* untuk Nama.
3. *Input Field* untuk *Password*.
4. *Input Field* untuk Ulangi *Password*.
5. *Input Field* untuk memilih provinsi asal.
6. Tombol untuk register / daftar.
7. Tombol untuk kembali ke halaman Login.

### Mengakses Halaman Penjelasan Soal

Berikut ini akan dijelaskan cara pengguna mengakses halaman Penjelasan Soal:

1. Setelah *login* maka pengguna akan diarahkan pada halaman Penjelasan Soal.
2. Halaman Penjelasan Soal berisi teks penjelasan dan tombol untuk lanjut mengerjakan soal seperti pada Gambar A.7.7.



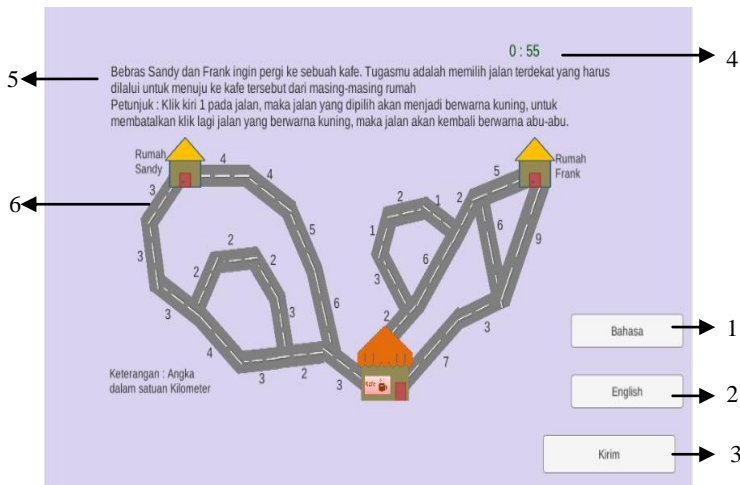
**Gambar A.7.7 Halaman Penjelasan Soal**

1. Teks penjelasan soal.
2. Tombol untuk mulai mengerjakan soal.

## Mengakses Halaman Soal

Berikut ini akan dijelaskan cara pengguna mengakses halaman Soal beserta beberapa tombol yang ada pada halaman Soal:

1. Setelah peserta mengakses halaman Penjelasan Soal lalu pengguna memilih tombol Mulai. Setelah itu akan muncul soal yang harus dikerjakan oleh pengguna seperti pada Gambar A.7.8.
2. Penulis memilih salah satu dari empat soal sebagai contoh pada lampiran ini.



**Gambar A.7.8 Halaman Soal**



1. Tombol aksi untuk menerjemahkan soal dalam bahasa Indonesia.
2. Tombol aksi untuk menerjemahkan soal dalam bahasa Inggris.
3. Tombol aksi untuk menuju halaman soal berikutnya.
4. Teks untuk hitung mundur waktu soal.
5. Teks soal.
6. Tombol untuk memilih jalan.

### Mengakses Halaman Selesai

Berikut ini akan dijelaskan cara pengguna mengakses halaman Selesai:

1. Setelah peserta mengerjakan soal sejumlah empat soal, maka peserta akan diarahkan menuju halaman Selesai.
2. Halaman selesai menampilkan nilai total yang didapat oleh peserta, tombol RANKING dan tombol Log Out yang dapat dilihat pada Gambar A.7.9.



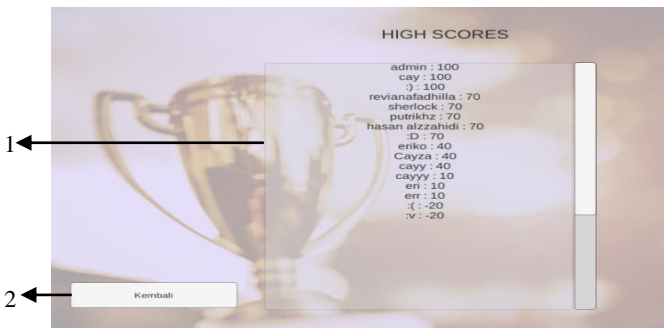
**Gambar A.7.9 Halaman Selesai**

1. Teks untuk menampilkan total nilai yang didapat oleh peserta.
2. Tombol untuk menuju halaman Peringkat.
3. Tombol untuk *log out* dan menuju halaman Beranda.

### Mengakses Halaman Peringkat

Berikut ini akan dijelaskan cara pengguna mengakses halaman Peringkat:

1. Halaman peringkat akan muncul ketika peserta memilih tombol RANKING pada halaman Selesai.
2. Halaman peringkat berisi *username* dan total nilai yang terurut dari tertinggi hingga terendah yang dapat dilihat pada Gambar A.7.10.



**Gambar A.7.10 Halaman Peringkat**

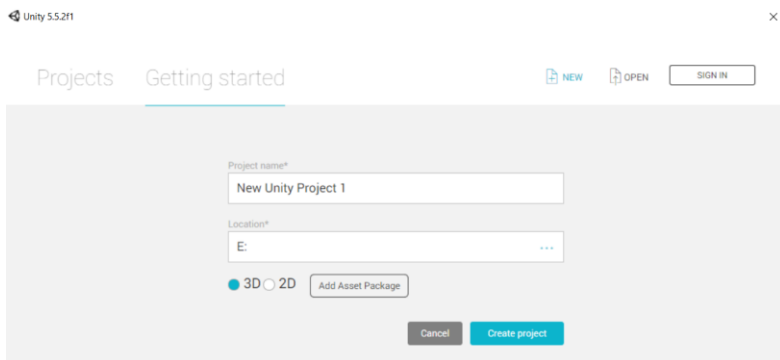
1. Teks untuk menampilkan peringkat peserta dari nilai tertinggi ke terendah.
2. Tombol untuk kembali ke halaman Selesai.

## LAMPIRAN B

### PANDUAN PEMBUATAN SOAL

Berikut ini adalah lampiran untuk urutan cara pembuatan soal pada Unity. Pada bab ini, penulis lampirkan cara membuat satu soal dari dua soal yang dibuat oleh penulis.

#### Pembuatan *Project* di Unity



**Gambar B.8.1 Tampilan Awal Unity**

1. Buat *project* baru pada Unity dengan cara membuka aplikasi Unity, lalu klik tulisan “NEW” maka akan muncul jendela seperti Gambar B.8.1 di atas.
2. Terdapat dua *fields*, yaitu nama *project* dan lokasi penyimpanan *project*. Pembuat soal dapat memilih dengan bebas nama *project* dan lokasi penyimpanan.
3. Terdapat pilihan “3D” atau “2D”. Jika pembuat soal memilih “3D” maka *project* akan dibuat dalam bentuk 3 dimensi. Penulis menyarankan untuk membuat project dalam 3 dimensi karena pada *project* 3 dimensi dapat diubah menjadi 2 dimensi.

### **Pembuatan Latar di *Project***

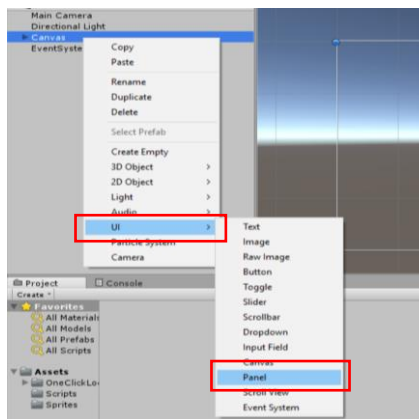
1. Klik kanan pada *hierarchy* untuk membuat *canvas*, *canvas* ini diperlukan untuk menjadi latar.
2. Klik kanan pada *canvas* untuk membuat panel seperti pada Gambar B.8.2.
3. Pilih warna yang diinginkan dengan cara memilih baris “Color” pada panel seperti ditunjukkan pada Gambar B.8.3
4. Setelah memilih warna, panel akan berubah warna seperti pada Gambar B.8.4 setelah itu, klik tombol “x” berwarna merah untuk selesai.

### **Pembuatan Soal 1**

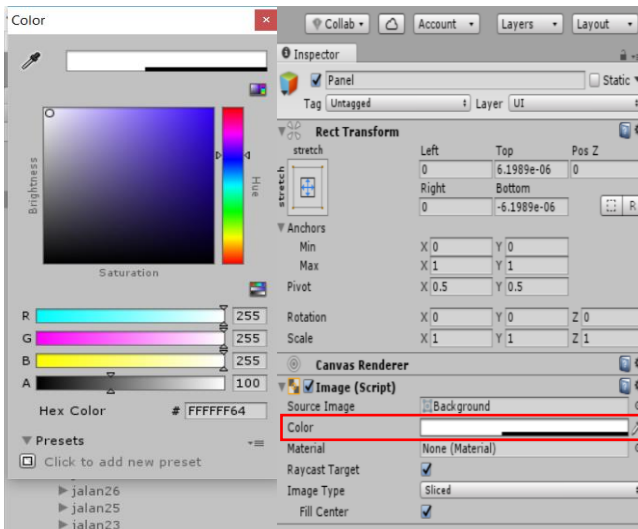
1. Buat teks untuk penjelasan soal dengan cara klik kanan pada panel lalu pilih menu “UI” dan pilih “Text” seperti pada gambar B.8.5
2. Tulis kalimat yang diinginkan di *box text* pada *text script*, hasil implementasi *text* soal dapat dilihat pada Gambar B.8.6.
3. Masukkan gambar dari simpanan di *file* tempat menyimpan gambar seperti yang dapat dilihat pada Gambar B.8.7
4. Masukkan gambar ke dalam Unity dengan cara *drag and drop* gambar dari *file* ke dalam folder *sprites*.
5. Ubah tipe gambar menjadi 2D (*sprites and UI*).
6. Buat tombol aksi dengan cara klik kanan pada *canvas* lalu pilih menu “UI” dan pilih “Button” seperti pada Gambar B.8.8.
7. Klik kanan pada panel untuk membuat gambar, lalu pilih menu “UI” dan pilih “Image” seperti pada Gambar B.8.9.
8. Drag and drop gambar yang telah dimasukkan sebelumnya ke dalam “Source Image” seperti pada

Gambar B.8.10, hasil implementasi gambar dapat dilihat pada Gambar B.8.11.

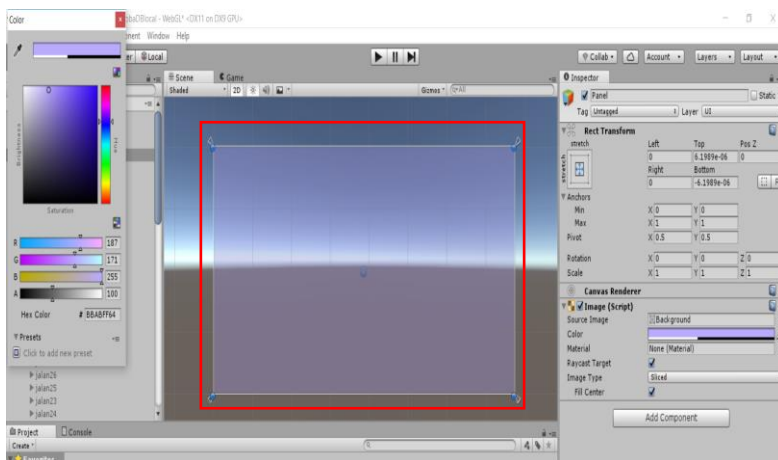
9. Gambar tombol diubah menjadi tombol yang bergambar jalan
10. Klik kanan pada panel untuk membuat tombol, lalu pilih menu “UI” dan pilih “Button” seperti pada Gambar B.8.12.
11. Buat gambar seperti pada tahap membuat gambar sebelumnya kemudian *drag and drop* gambar ke dalam “Source Image” seperti pada Gambar B.8.13, hasil implementasi tombol dapat dilihat pada Gambar B.8.14.
12. Berilah *text* pada setiap jalan dengan cara seperti tahap pembuatan *text* soal sebelumnya, hasil implementasi *text* pada jalan dapat dilihat pada Gambar B.8.15.
13. Berilah keterangan Rumah Sandy, Rumah Frank, dan kilometer dengan cara seperti pada tahap pembuatan *text*.
14. Berilah tombol yang berfungsi untuk mengirim jawaban dengan cara seperti pada tahap membuat tombol sebelumnya.
15. Hasil implementasi soal 1 secara keseluruhan dapat dilihat pada Gambar B.8.16.



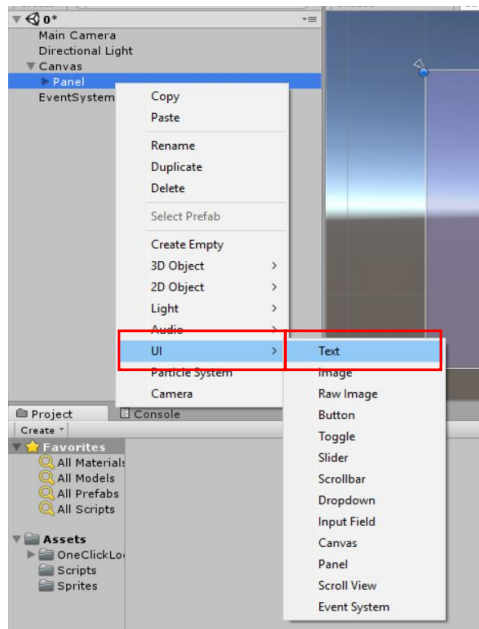
**Gambar B.8.2 Membuat Panel**



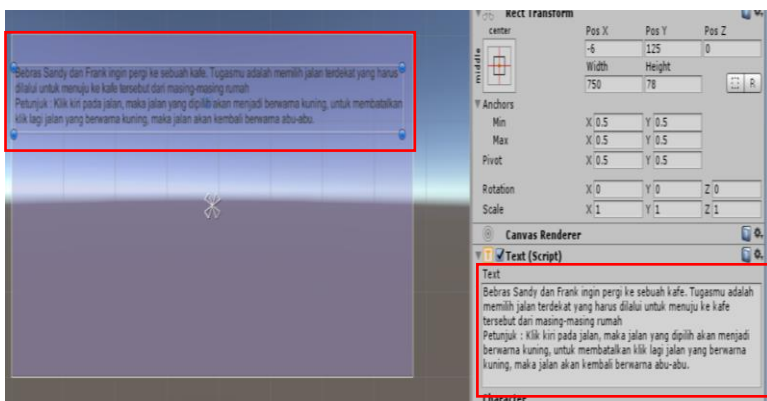
**Gambar B.8.3 Memberi Warna pada Panel**



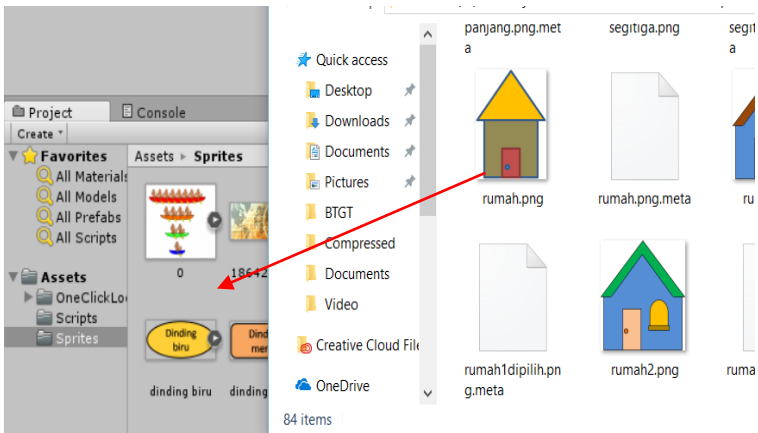
**Gambar B.8.4 Hasil Panel yang Telah Diberi Warna**



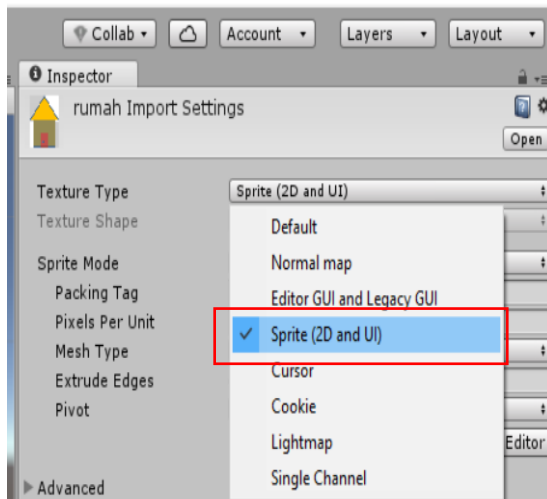
**Gambar B.8.5 Membuat *Text***



**Gambar B.8.6 Implementasi Pembuatan *Text***

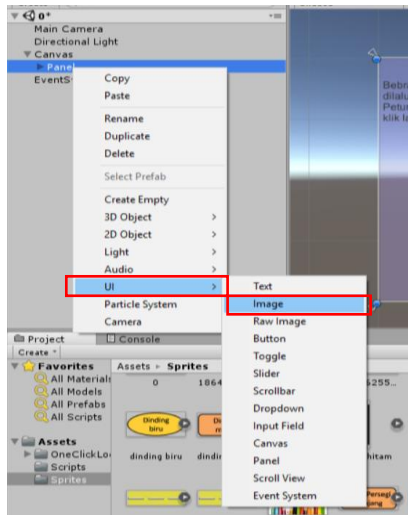


**Gambar B.8.7 Input Gambar**

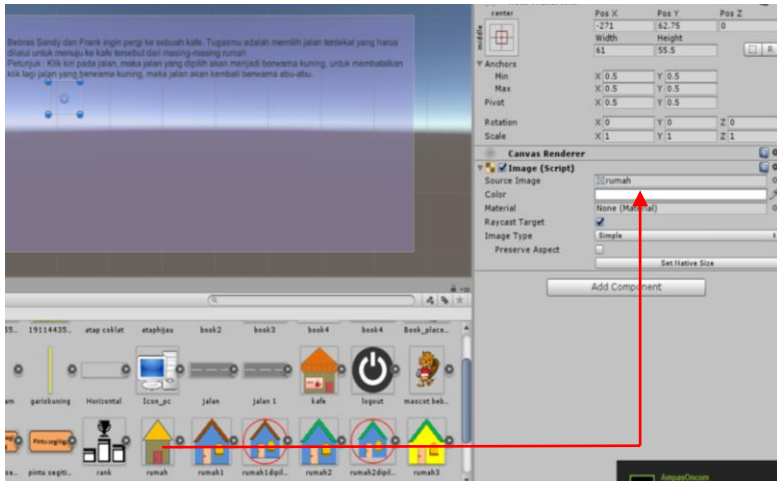


**Gambar B.8.8 Tipe Gambar**





**Gambar B.8.9 Membuat Gambar**

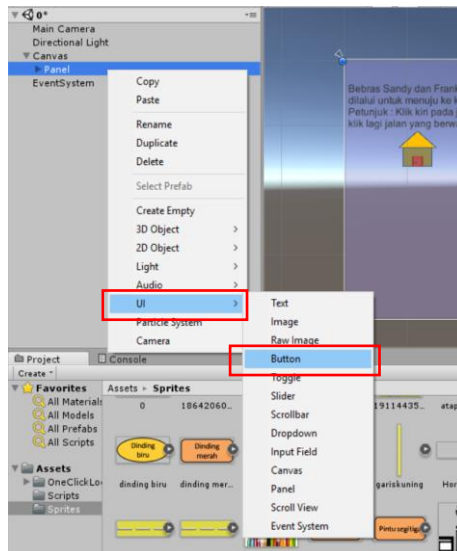


**Gambar B.8.10 Drag and Drop Gambar**

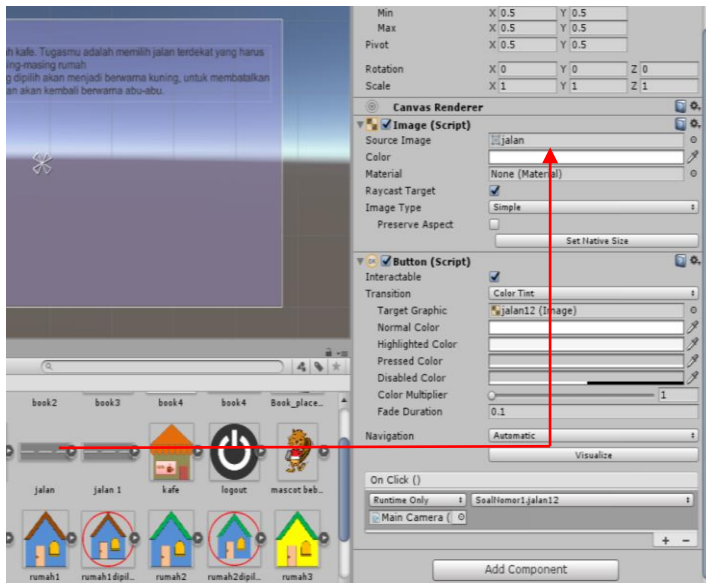
(11 on DX9 GPU>



**Gambar B.8.11 Implementasi Pembuatan Gambar**



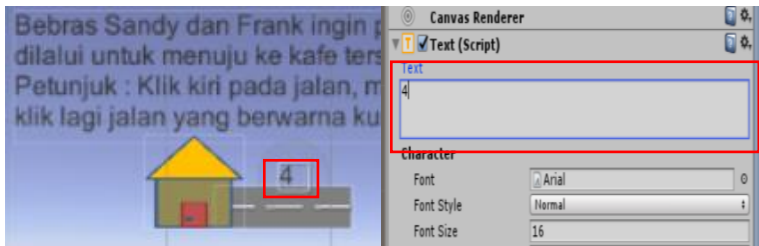
**Gambar B.8.12 Membuat Tombol**



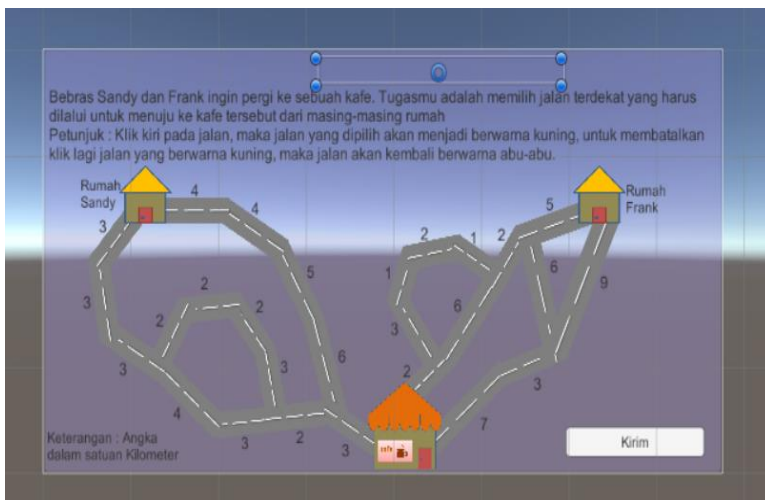
**Gambar B.8.13 Implementasi Tombol**



**Gambar B.8.14 Hasil Implementasi Tombol**



**Gambar B.8.15 Implementasi *Text* pada Jalan**

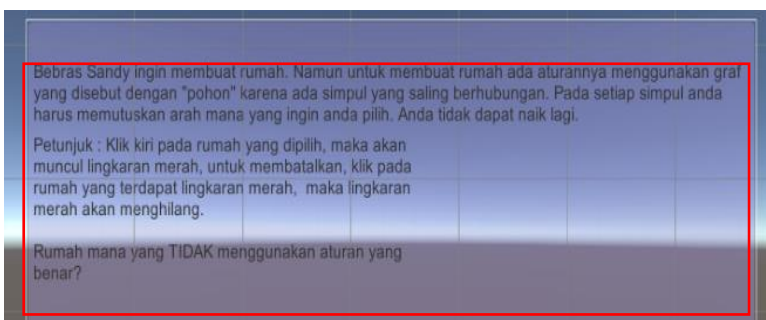


**Gambar B.8.16 Hasil Implementasi Soal 1 Secara Keseluruhan**

## Pembuatan Soal 2

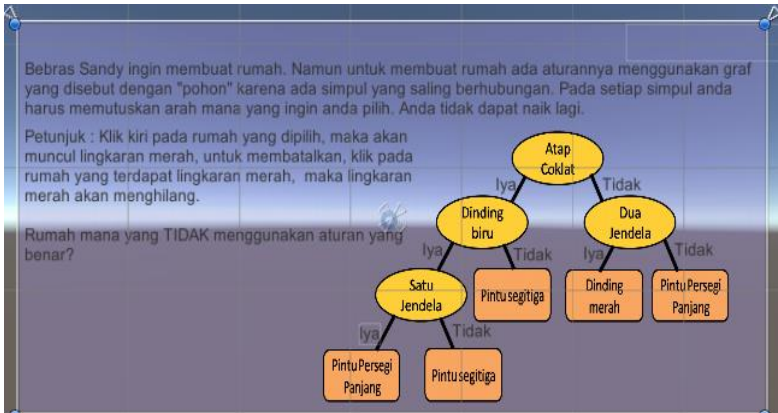
1. Buat teks seperti pada tahap membuat teks sebelumnya di soal 1. Tulis kalimat yang diinginkan di *box text* pada *text script*.
2. Hasil implementasi *text* soal dapat dilihat pada Gambar B.8.17.

3. Buat gambar seperti pada tahap membuat gambar sebelumnya di soal 1.
4. Hasil implementasi gambar sisi dan simpul dapat dilihat pada Gambar B.8.18.
5. Berilah *text* pada setiap sisi dengan cara seperti tahap pembuatan *text* soal 1 sebelumnya.
6. Hasil implementasi *text* pada sisi dapat dilihat pada Gambar B.8.19.
7. Hasil Implementasi semua simpul, sisi, dan teks dapat dilihat pada Gambar B.8.20.
8. Gambar tombol diubah menjadi tombol yang bergambar rumah.
9. Buat tombol seperti pada tahap membuat tombol sebelumnya di soal 1.
10. Buat gambar seperti pada tahap membuat gambar sebelumnya di soal 1, kemudian drag and drop gambar ke dalam “Source Image” seperti pada Gambar B.8.21.
11. Hasil implementasi tombol dapat dilihat pada Gambar B.8.22.
12. Berilah tombol yang berfungsi untuk mengirim jawaban dengan cara seperti pada tahap membuat tombol sebelumnya di soal 1.
13. Hasil Implementasi soal 2 secara keseluruhan dapat dilihat pada Gambar B.8.23.

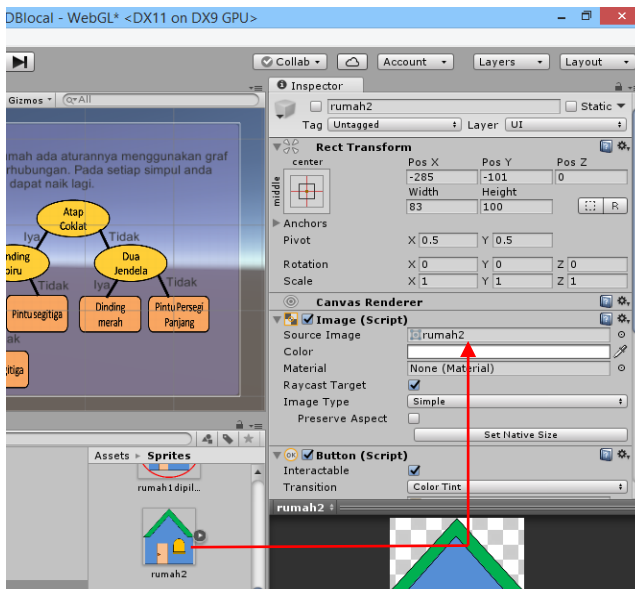


**Gambar B.8.17 Hasil Implementasi Pembuatan Teks**





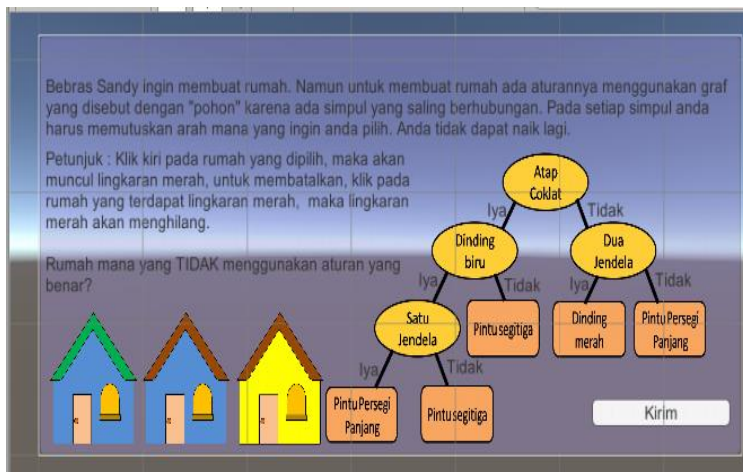
**Gambar B.8.20 Hasil Implementasi Gambar Semua Simpul, Sisi, dan Teks pada Sisi**



**Gambar B.8.21 Implementasi Pembuatan Tombol**



**Gambar B.8.22 Hasil Implementasi Pembuatan Tombol**



**Gambar B.8.23 Hasil Implementasi Soal 2 Secara Keseluruhan**



### Pembuatan Soal 3

1. Pada pembuatan soal 3 memiliki tahap-tahap yang sama seperti pembuatan soal 1. Pada soal 3 hanya memiliki perbedaan bentuk serta jumlah pada implementasi teks, tombol, dan gambar dari soal 1.
2. Hasil implementasi soal 3 secara keseluruhan dapat dilihat pada Gambar B.8.24.

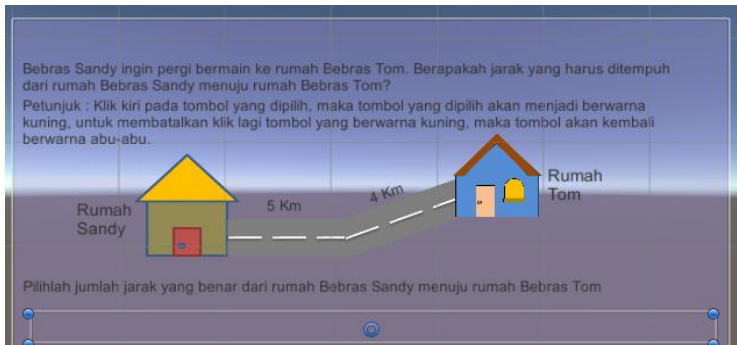


**Gambar B.8.24 Hasil Implementasi Soal 3 Secara Keseluruhan**

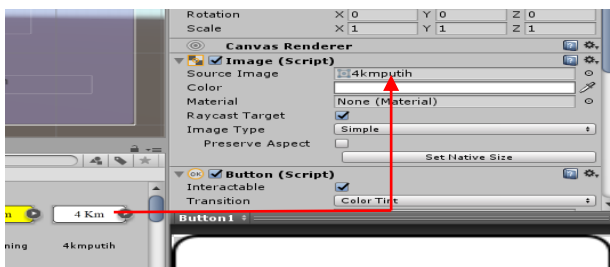
### Pembuatan Soal 4

1. Buat teks seperti pada tahap membuat teks sebelumnya di soal 1. Tulis kalimat yang diinginkan di *box text* pada *text script*.
2. Buat gambar seperti pada tahap membuat gambar sebelumnya di soal 1. Drag and drop gambar yang telah dimasukkan sebelumnya ke dalam “Source Image”.
3. Hasil implementasi *text* dan gambar dapat dilihat pada Gambar B.8.25.
4. Gambar tombol diubah menjadi tombol yang bergambar satuan kilometer.

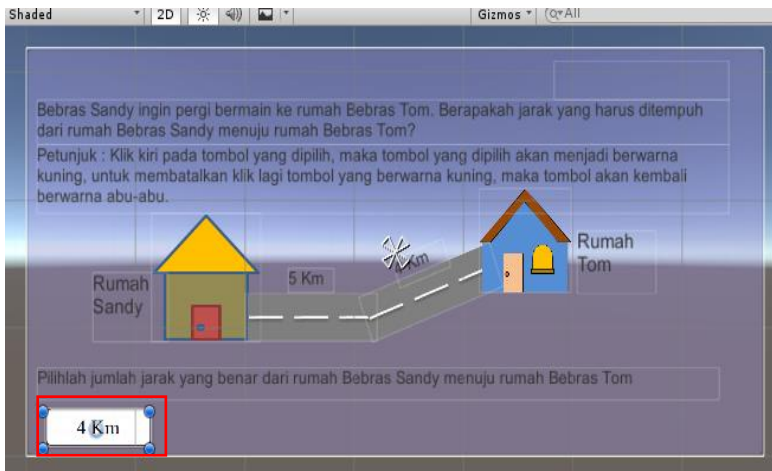
5. Buat tombol seperti pada tahap membuat tombol sebelumnya di soal 1.
6. Buat gambar seperti pada tahap membuat gambar sebelumnya di soal 1, kemudian drag and drop gambar ke dalam “Source Image” seperti pada Gambar B.8.26.
7. Hasil implementasi tombol dapat dilihat pada Gambar B.8.27.
8. Berilah tombol yang berfungsi untuk mengirim jawaban dengan cara seperti pada tahap membuat tombol sebelumnya di soal 1.
9. Hasil Implementasi soal 4 secara keseluruhan dapat dilihat pada Gambar B.8.28.



**Gambar B.8.25 Hasil Implementasi Teks dan Gambar**



**Gambar B.8.26 Implementasi Tombol**



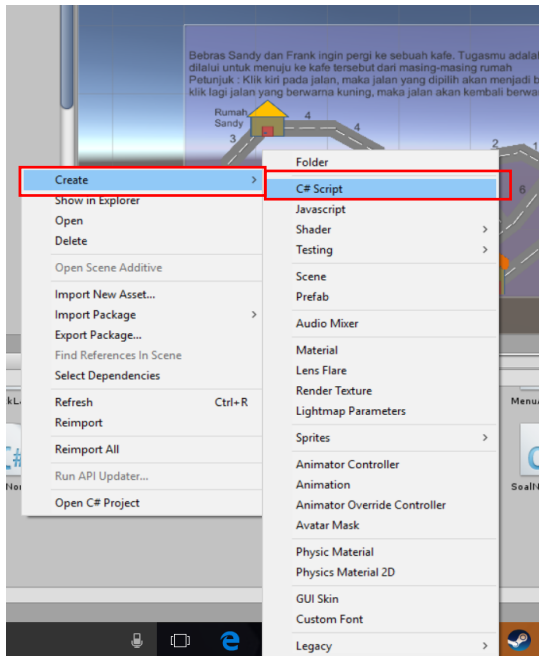
**Gambar B.8.27 Hasil Implementasi Tombol**



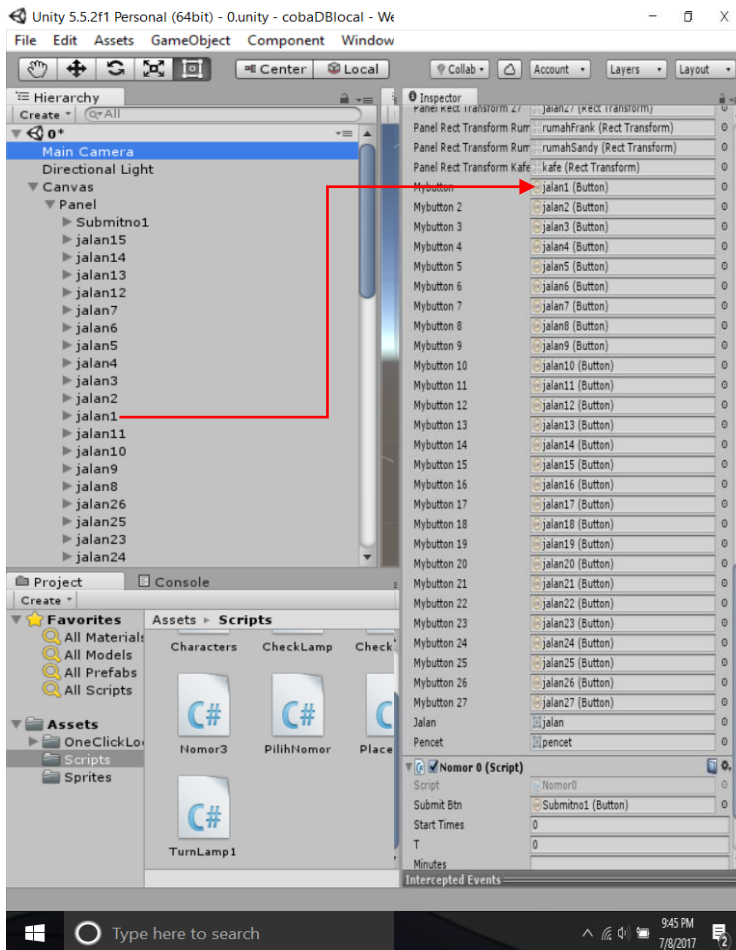
**Gambar B.8.28 Hasil Implementasi Soal 4 Secara Keseluruhan**

### Pembuatan *Script*

1. Klik kanan pada bagian *Project* di bawah, klik kanan lalu pilih “Create”, lalu pilih C# *Script* seperti pada Gambar B.8.15.
2. Untuk mengimplementasikan *Script*, pembuat soal harus *drag and drop script* menuju Main Camera
3. Jika pada *Script* diharuskan melakukan *reference* pada objek tertentu maka akan muncul baris-baris seperti pada Gambar B.8.17. Sebagai contoh, penulis menginisialisasi *button* “jalan1”. Maka pembuat soal harus *drag button* “jalan1” pada *hierarchy* menuju pada baris-baris pada *Script*.



Gambar B.8.29 Pembuatan *Script*



**Gambar B.8.30 Reference Object pada Script**

## Pengelolaan File

Pengelolaan file dapat dilakukan dengan cara masuk link “[bebrasindonesia.com/cpanel](http://bebrasindonesia.com/cpanel)”, kemudian pilih “Pengelola File”, setelah itu pilih “Unggah”. Pilihlah file yang ingin diunggah.

*[Halaman ini sengaja dikosongkan]*

## LAMPIRAN C KUISONER PESERTA



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**KUISONER TUGAS AKHIR – 5113100122 ERIKO AGNITA MASROERI**

---

**RANCANG BANGUN APLIKASI EDUKASI INTERAKTIF UNTUK MELATIH BERPIKIR KOMPUTASIONAL  
MATERI GRAF**

---

**Identitas Responden**

Nama Lengkap : Putri Khumairotus Zehra      Usia : 17 tahun

Pekerjaan : Pelajar      Jenis Kelamin : L/P

**A. PENILAIAN TERHADAP APLIKASI**

Isilah tabel di bawah dengan menggunakan tanda (v)

SS = Sangat Setuju      S = Setuju      C = Cukup  
TS = Tidak Setuju      STS = Sangat Tidak Setuju

No	Parameter Antarmuka	STS	TS	C	S	SS
1	Aplikasi memiliki tampilan dan desain yang menarik				✓	
2	Aplikasi memiliki menu yang mudah digunakan				✓	
3	Aplikasi memiliki tata letak tombol yang sesuai				✓	
	Parameter Performa dan Kenyamanan					
4	Aplikasi nyaman untuk digunakan				✓	
5	Aplikasi di akses dengan cepat					✓
	Parameter Materi Soal					
5	Aplikasi memberikan soal yang menarik dan menantang					✓
6	Aplikasi memberikan soal yang sesuai dengan materi graf				✓	

**B. KRITIK DAN SARAN**

Penggunaan gambar lingkaran ketika muncul intuksi soal dan cara mengerjakan selang-seling harus diperbaiki lebih banyak. Jika perlu ditambahkan petunjuk seperti sebelum masuk ke nomor selanjutnya harus dicek dulu. Ada beberapa soal yang harus diperbaiki tampilannya. ( baik ukuran dan gambar )

Surabaya, 30 Juni 2017

Putri

**Gambar C.9.1 Kuisoner Putri**



KUISIONER TUGAS AKHIR – 5113100122 ERIKO AGNITA MASROERI

**RANCANG BANGUN APLIKASI EDUKASI INTERAKTIF UNTUK MELATIH BERPIKIR KOMPUTASIONAL**  
**MATERI GRAF**

**Identitas Responden**

Nama Lengkap : M. Hasanudin Zubdi Usia : 15 th  
Pekerjaan : pelajar Jenis Kelamin : (L) P

**A. PENILAIAN TERHADAP APLIKASI**

Isilah tabel dibawah dengan menggunakan tanda (v)

SS = Sangat Setuju

S = Setuju

C = Cukup

TS = Tidak Setuju

STS = Sangat Tidak Setuju

No	Parameter Antarmuka	STS	TS	C	S	SS
1	Aplikasi memiliki tampilan desain yang menarik				✓	
2	Aplikasi memiliki menu yang mudah digunakan				✓	
3	Aplikasi memiliki tata letak tombol yang sesuai			✓		
	Parameter Performa dan Kenyamanan					
4	Aplikasi nyaman untuk digunakan				✓	
5	Aplikasi diakses dengan cepat			✓		
	Parameter Materi Soal					
5	Aplikasi memberikan soal yang menarik dan menantang					✓
6	Aplikasi memberikan soal yang sesuai dengan materi graf					✓

**B. KRITIK DAN SARAN**

Pengaturan dan penilaian skor lebih baik dapat diakui  
dan kalau soal diberi penjelasan yg lebih

Surabaya, 30 Jan 2017

LP  
M. Hasanudin Zubdi

**Gambar C.9.2 Kuisioner Hasanudin**





KUISIONER TUGAS AKHIR – 5113100122 ERIKO AGNITA MASROERI

**RANCANG BANGUN APLIKASI EDUKASI INTERAKTIF UNTUK MELATIH BERPIKIR KOMPUTASIONAL  
MATERI GRAF**

**Identitas Responden**

Nama Lengkap : REVIANA FADHILLA C. Usia : 15 tahun  
Pekerjaan : Pelajar Jenis Kelamin : t / P

**A. PENILAIAN TERHADAP APLIKASI**

Isilah tabel dibawah dengan menggunakan tanda (v)

SS = Sangat Setuju

S = Setuju

C = Cukup

TS = Tidak Setuju

STS = Sangat Tidak Setuju

No	Parameter Antarmuka	STS	TS	C	S	SS
1	Aplikasi memiliki tampilan desain yang menarik				✓	
2	Aplikasi memiliki menu yang mudah digunakan					✓
3	Aplikasi memiliki tata letak tombol yang sesuai				✓	
	Parameter Performa dan Kenyamanan					
4	Aplikasi nyaman untuk digunakan				✓	
5	Aplikasi diakses dengan cepat					✓
	Parameter Materi Soal					
5	Aplikasi memberikan soal yang menarik dan menantang					✓
6	Aplikasi memberikan soal yang sesuai dengan materi graf				✓	

**B. KRITIK DAN SARAN**

Soal terlalu panjang dan agak susah dipahami  
Gambar lebih dibuat menarik selebihnya menantang

Surabaya, 27 Juni 2017

Reviana Fadhilla

**Gambar C.9.3 Kuisoner Reviana**

**Tabel C.9.1 Hasil Kuisioner Peserta**

No	Pertanyaan	Jumlah				
		STS	TS	C	S	SS
1	Aplikasi memiliki tampilan dan desain yang menarik	0	0	0	3	0
2	Aplikasi memiliki menu yang mudah digunakan	0	0	0	2	1
3	Aplikasi Memiliki tata letak tombol yang sesuai	0	0	1	2	0
4	Aplikasi nyaman untuk digunakan	0	0	0	3	0
5	Aplikasi diakses dengan cepat	0	0	1	0	2
6	Aplikasi memberikan soal yang menarik dan menantang	0	0	0	0	3
7	Aplikasi memberikan soal yang sesuai dengan materi G	0	0	0	2	1

Keterangan:

STS : Sangat Tidak Setuju

TS : Tidak Setuju

C : Cukup

S : Setuju

SS : Sangat Setuju

## **BIODATA PENULIS**



Eriko Agnita Masroeri dilahirkan pada tanggal 24 Januari 1995 di Surabaya. Penulis menempuh pendidikan di SD Muhammadiyah 4 Surabaya, SMPN 19 Surabaya, SMAN 17 Surabaya, dan melanjutkan S1 Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember, Surabaya.

Memiliki pengalaman organisasi sebagai Staf Departemen Kewirausahaan dan Minat Bakat Himpunan Teknik Computer-Informatika 2014-2015 dan Staf Ahli Departemen Kewirausahaan Himpunan Teknik Computer-Informatika 2015-2016. Selain itu, juga memiliki pengalaman kepanitiaan, diantaranya sebagai Staf kesekretariatan SCHEMATICS 2014 dan Wakil biro Dana dan Usaha SCHEMATICS 2015.